Quantitative Analysis of GAN samples

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper we quantitatively evalute samples produced with adversarial methods, specially Generative Adversarial Networks. We analyze summary statistics of real and fake data and compute distances based on the marginal distribution of perceptually significant features. We provide results on image, music and speech data and show that GAN generated samples have signatures that can be used to detect adversarial attacks.

1 Introduction

Since the first Generative Adversarial Networks paper in 2014, there have been many advances and publications related to the topic, including theoretical research on the framework, such as LSGAN, WGAN, Improved WGAN, Mixed GANs, Began, Energy Gans..., Although at the beginning GAN research was focused on natural images, it has been expanding to language models (ref iwgan) and music (ref midi gan and my paper).

Unlike variational auto encoders and other methods, most of the evaluation of the output of Generators trained with the GAN framework is qualitative. When comparing to other GAN methods, authors usually list generation of higher quality sample as one of the advantages of their method. Interestingly, although decrease in loss can be correlated with increase in image quality, this is not always the case and authors still relly on visual inspection of generated images.

Based on visual inspection, authors confirm that they have not observed mode collapse or that their framework is robust to mode collapse if some criteria is meet. In practice, there is abundance of github issues where practicioners report mode collapse. Theoretically, this is dissonant with research evaluating mode collapse and variety in samples generated with the GAN framework.

In the early GAN papers, authors used quantititave measures to evaluate GAN samples but the methods used perform poorly in high dimensional spaces or have high variance. Evaluating GAN samples quantitatively is hard because it depends on the existence of perceptually meaningful features. For example, consider the generation of images of mamals: although it is possible to compute how far the color histogram of the fake samples are from the real samples, we do not yet have robust algorithms able to verify if an image follows specifications or properties derived from anatomy.

Facing this challenge, there is a research trend that uses features computed over the real samples, e.g. label statistics, for training and evaluating generative models. This is a promissing field and we foresee that it will benefit considerably from developments in visual question answering.

This paper is related to this research trend and quantitatively evaluates GAN generated samples by marginalizing perceptually meaningful features and computing the distance between the distribution of these features in the real and the fake[1] data. The intuition is that as the number of distribution of features being compared increases, the more likely it is that the combination of these features is a meaningful representation of the true data. We offer the following contributions in this paper:

[1] Data sampled from the generator

- We provide an alternative method to evaluate GAN samples manually
- We show that GAN samples have universal signatures
- We show that GAN are note able to fully approximate simple distributions
- We show significant differences exist between the marginal distribution of features from the real and fake data
- We compare the real distribution with adversarial data generated using the fast gradient sign method

2 Related work

In the past few years, several publications have investigated the use of the Generative Adversarial Networks framework for generation of samples and unsupervised feature learning. Following the practice started in paper and followed in Goodfellow et al ground-breaking GAN paper, earlier papers evaluate the quality of the generator by fitting a Gaussian Parzen window[2] to the GAN samples and reporting the log-likelihood of the test set data under this distribution. It is know that this method has some drawbacks, including its high variance and bad performance in high dimensional spaces.

In their brilliant publications, LSGAN, WGAN and IWGAN propose alternative objective functions and algorithms that circunvemt problems that are common when using the original GAN objective, which minimizes the Jenson-Shannon Divergence. The problems addressed include instability of learning, mode collapse and meaningful learning curves.

These alternatives do not eliminate the need of visual inspection of samples. Although visual inspection can be emotionally pleasing, it can be extremelly cumbersome[3] and it does not provide a clear description of the numerical properties of the generated samples, nor the diversity of the generator's output. In BEGAN, the authors propose a solution to the diversity problem by introducing a new hyper-parameter γ with a loss derived from the Wasserstein distance. Naturally, this new hyper-parameter does not target the diverstiy of a specific attribute of the images and the results in the paper suggest that in their experiments γ is also correlated with the variety of the color pallete.

Related to our paper, work by Deepak shows a very interesting approach, where summary statistics of the output label are used to train the generator and evaluate its output. In his paper, Deepak proposes a method that uses a novel loss function to optimize for any set of linear constraints on the output space of a CNN. In our paper, we draw inspiration from formal methods and specification mining. We approach such constraints as specifications that are mined from the real data. For example, one extract specifications from anatomy to evaluate or target sample generation with GANs. We use the learned specifications to validate the output of the samples generated with the GAN framework on MNIST images,estimate speech and music.

3 Method

In this section we describe our method in detail. We start by describing the hypothesis we will evaluate in our paper using MNIST, music and speech data.

3.1 Hypotheses

Hypothesis 1 (H1): *Generative models can approximate the distribution of real data and hallucinate fake data that has some variaety and resembles the real data*

Although this hypothesis is trivial for experiments that have already been conducted, it is the first condition for our experiments with music and speech data. To our knowledge there are no publications where GANs are successful in hallucinating music and speech data. During out experiments we prove that this hypothesis is true.

Hypothesis 2 (H2): *The real data has useful properties that can be extracted computationally.*

[2] Kernel Density Estimation

[3] I'll never train GANs again

By useful we refer to properties that are closely related to the real data itself. For example, computing the distribution MNIST pixel values might be not useful for assessing drawing quality. However, it might be useful to evaluate if a random MNIST samples is real or fake data.

Hypothesis 3 (H3): *The fake data has properties that are hardly noticed with non-computational inspection.*

Visual inspection of generated samples has become the norm for the evaluation of samples generated using the GAN framework. We investigate if there are properties common to all GAN samples or properties that significantly differ between the real data and the fake data. This hypothesis supports the next hypothesis related to adversarial attacks.

Hypothesis 4 (H4): *The difference in properties can be used to identify the source (real or fake)*

The development of generative models for digital media announce the iminent rise of adversarial attacks. We investigate if these differences can be used to detect if the data was generated with the GAN framework or is an adversarial attack.

We call the reader's attention that approximating the distribution over features computed on the real data does not guarantee that the real data is being approximated. Formally speaking: Consider $X \sim Z$, i.e. X distributed as Z, and $f(X) \sim W$, where $f : X \mapsto Y$. If $A \sim B$ and B approximates Z, then $f(A) \sim D$ must also approximate W. However, a distribution that approximates W is not guaranteed to approximate Z.

3.2 Learning properties

In this subsection, we describe the properties that we mine from data. They comprise of properties that are perceptually related with the image and properties that are not perceptually related but that can be used to identify the source of the image. Consider the single channel image I with dimensions R by C, where $I_{r,c}$ is the pixel intensity of the pixel at row r and column c

3.2.1 Summary Statistics

Consists of the distribution of mean, standard deviation, kurtosis and skewness feature values over all images. It is applied to pixel intentisy and some features described below.

3.2.2 Spectral Moments

The spectral centroid is a feature commonly used in the audio domain, where it represents the barycenter of the spetrum. Given an image, for each column we transform the pixel values into probabilities by normalizing them by the column sum, after which we take the expected row value. Given one image column, we define r as the pixel intensity at row r, and

$$p(r) = \frac{r}{\sum_{r \in R} r} \tag{1}$$

From these definitions, it immediately follows that the first, second, third and fourth moments can be described as follows:

$$\mu = \int r p(r) \partial r \tag{2}$$

$$\sigma^2 = \int (r - \mu)^2 p(r) dr \tag{3}$$

$$\gamma_1 = \frac{\int (r - \mu)^3 p(r) dr}{\sigma^3} \tag{4}$$

$$\gamma_2 = \frac{\int (r - \mu)^4 p(r) dr}{\sigma^4} \tag{5}$$

Figure 1 shows the spectral centroid computed on sample of MNIST training data.

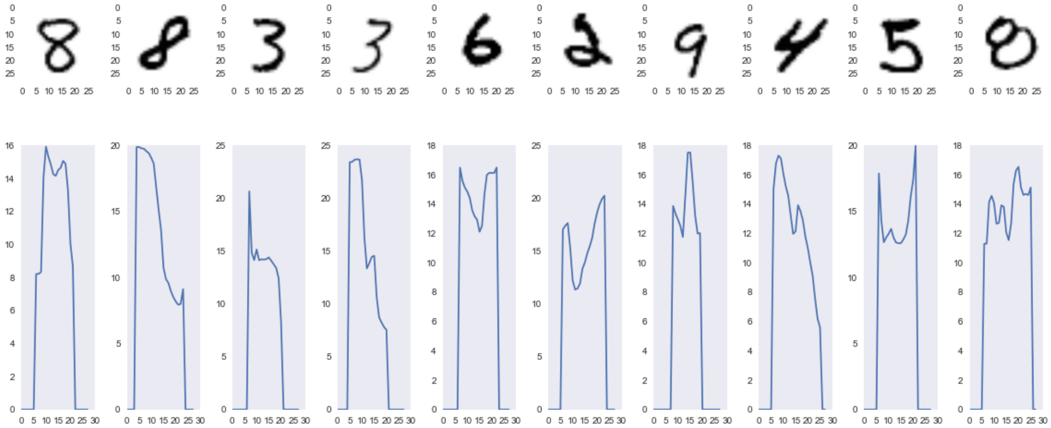

Figure 1:

3.2.3 Spectral Slope

Is computed by linear regression on the sepectral centroid with window of size 7. Figure 2 shows these features computed on a sample of MNIST training data.

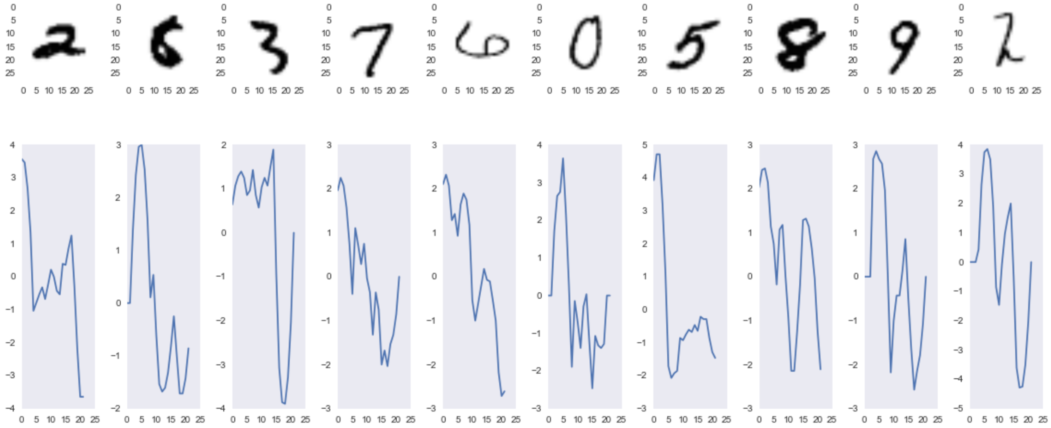

Figure 2:

3.2.4 Transition Matrix

Transition matrix is computed only for chromagram representations of piano rolls.
Equation

3.3 Distance Measures

We use the Kolgomorov-Smirnov Two Samples Test
Equation

We use the Jensen-Shannong Divergence
Equation

3.4 Generative Adversarial Networks

We investigate the DCGAN architecture under LSGAN, WGAN, IWGAN objective functions.

4 Experiments

4.1 MNIST

We compare the the distribution of features computed over the MNIST train data to the distribution of the same features computed over MNIST test data, samples generated with LSGAN, improved WGAN, adversarial samples computed using the fast gradient sign method. The training data is scaled to $[-1, 1]$ and the scaled baseline is sampled from a binomial distribution with number of trials 1 and probability of success equal to the normalized mean value of pixel intentities in the MNIST training data, 0.13. The discriminator and the generator follow the DCGAN architecture. The classifier used to generate the adversarial samples is a three layer fully conected network with dropout on every layer (25%, 50%, 50%) and rectified linear units on the first and second layers. The ϵ parameter for the adversarial attack is set to .25(CHECK IF NOT .1). Following common practice in GAN training, each GAN model is trained until the loss plateaus and we are satisfied with the quality of the output.

Figure 3 shows samples drawn from MNIST train, test, LSGAN, IWGAN, FSGM and binomial respectively.

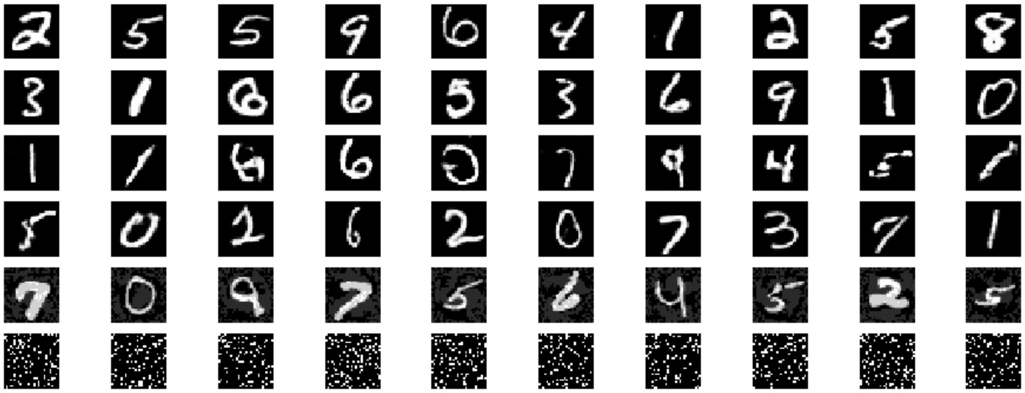

Figure 3:

Visual inspection of these generated samples can lead one to believe that IWGAN produces better samples than the LSGAN. Below we compare the distribution of pixel intensities of several data to MNIST's training data. Table 1 reveals that although the pixel intensities of LSGAN and IWGAN samples look similar to the training data, they are considerably different given the KS Two Sample Test[4]but not so different given the JSD. This can phaenomena can be understood by investigation of the empirical CDF of these samples in Figure 5. The pixel values of the samples generated with the GAN framework are mainly bimodal but distributed around -1 and 1. Such behavior will be present in any gradient descent method using an asymptotically converging non-linearity, such as sigmoid and tanh, immediately preceding the output of the generating function.

Table 1: KS Two Sample Test and JSD over the distribution of pixel values for different samples

	KS Two Sample Test		JSD
	Statistic	P-Value	
mnist_train	0.0	1.0	0.0
mnist_test	0.003177	8.501950e-35	2.955323e-05
mnist_lsgan	0.808119	0.0	0.013517
mnist_iwgan	0.701573	0.0	0.014662
mnist_adversarial	0.419338	0.0	0.581769
mnist_binomial	0.130855	0.0	0.0785009

Although this confirms our hypothesis 3 that there are properties that are hardly noticed with non-computational inspection, an adversary could easily apply thresholding to compensate for the

[4] Remember that the test statistic is inversely proportial to $\sqrt{n}$

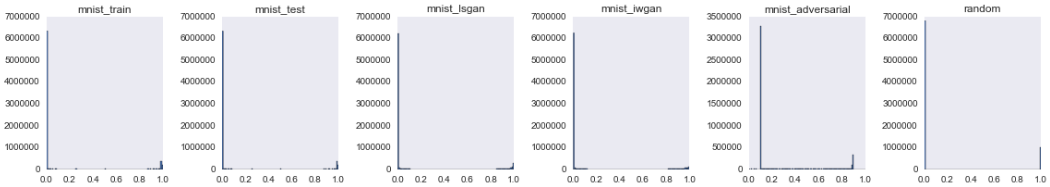

Figure 4:

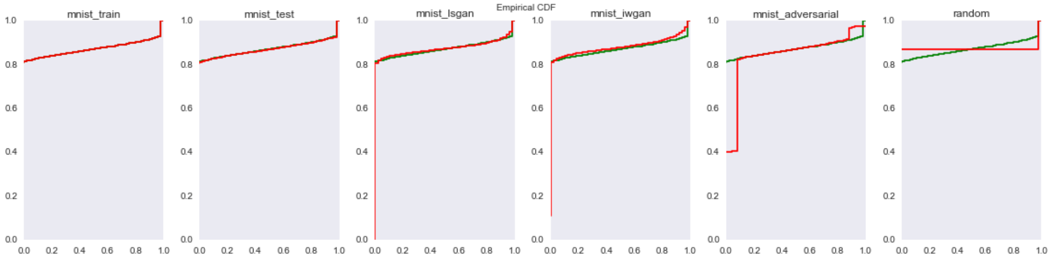

Figure 5:

asymptotically converging non-linearity such that the distribution of pixel values of fake samples become more similar to the real data.

For this reason, we use the distributions over other features on the real and fake data. Figure 6 shows that if on the one hand the distribution of slopes of the test data is similar to the training data, on the other hand this distribution on generated samples differs considerably from the training data, thus confirming hypothesis 4. In table 2 we show results of test statistics.

Table 2: KS Two Sample Test and JSD over the distribution of mean slope for different samples

	KS Two Sample Test		JSD
	Statistic	P-Value	
mnist_train	0.0	1.0	0.0
mnist_test	0.030699	0.000156	0.001872
mnist_lsgan	0.317200	0.0	0.177692
mnist_iwgan	0.478300	0.0	0.232894
mnist_adversarial	0.309099	0.0	0.022110
mnist_binomial	0.293200	0.0	0.084448

4.2 Bach Chorales

4.3 Speech

5 Conclusions

Acknowledgments

References

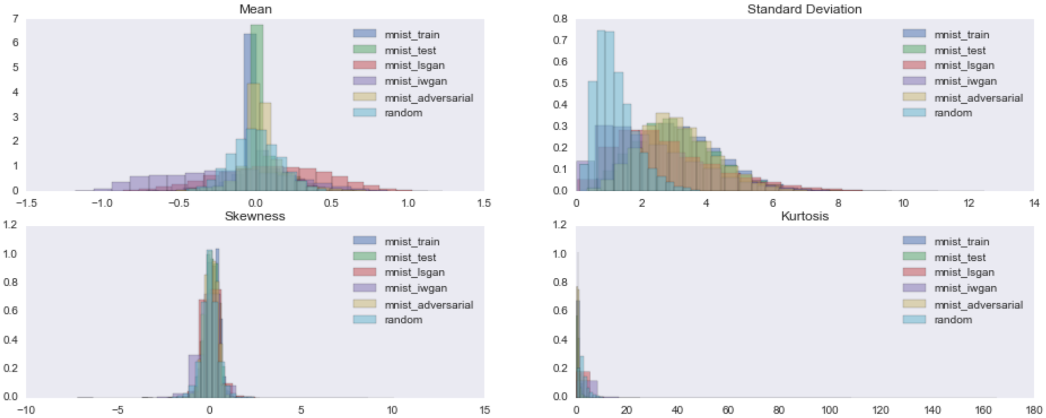
Mean
7
6
5
4
3
2
1
0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
mnist_train
mnist_test
mnist_lsgan
mnist_iwgan
mnist_adversarial
random
Standard Deviation
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2
4
6
8
10
12
14
mnist_train
mnist_test
mnist_lsgan
mnist_iwgan
mnist_adversarial
random
Skewness
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−10
−5
0
5
10
15
mnist_train
mnist_test
mnist_lsgan
mnist_iwgan
mnist_adversarial
random
Kurtosis
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80
100
120
140
160
180
mnist_train
mnist_test
mnist_lsgan
mnist_iwgan
mnist_adversarial
random

Figure 6: