Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2021) 29: 1524 – 1538

doi:10.3906/elk-2006-143

Determining overfitting and underfitting in generative adversarial networks using Fréchet distance

Enes EKEN*
Departmen of Electrical and Electronics Engineering, Faculty of Engineering, Aksaray University, Aksaray, Turkey

Received: 26.06.2020 • **Accepted/Published Online:** 27.10.2020 • **Final Version:** 31.05.2021

Abstract: Generative adversarial networks (GANs) can be used in a wide range of applications where drawing samples from a data probability distribution without explicitly representing it is essential. Unlike the deep convolutional neural networks (CNNs) trained for mapping an input to one of the multiple outputs, monitoring the overfitting and underfitting in GANs is not trivial since they are not classifying but generating a data. While training set and validation set accuracy give a direct sense of success in terms of overfitting and underfitting for CNNs during the training process, evaluating the GANs mainly depends on the visual inspection of the generated samples and generator/discriminator costs of the GANs. Unfortunately, visual inspection is far away of being objective and generator/discriminator costs are very nonintuitive. In this paper, a method was proposed for quantitatively determining the overfitting and underfitting in the GANs during the training process by calculating the approximate derivative of the Fréchet distance between generated data distribution and real data distribution unconditionally or conditioned on a specific class. Both of the distributions can be obtained from the distribution of the embedding in the discriminator network of the GAN. The method is independent of the design architecture and the cost function of the GAN and empirical results on MNIST and CIFAR-10 support the effectiveness of the proposed method.

Key words: Generative adversarial networks, Fréchet inception distance, overfitting, underfitting

1. Introduction

Classification of a multidimensional input data into one of the multiple classes and generating multidimensional output data from one of the multiple classes are the two attractive topics of machine learning. While the great successes at the classification tasks came after CNNs [1] and improved computer hardware, the most impressive results for generative models were obtained after GANs [2].

The GAN framework consists of two different neural networks, a discriminator D and a generator G competing with each other in an iterative game. During the training process, the generator receives a random input vector $\boldsymbol{z}$ sampled from $p_{\mathbf{z}}(\cdot)$ and generates a fake image $\hat{\mathbf{x}} = \mathrm{G}(\boldsymbol{z};\theta^G)$, where θ^G is the generator's parameters, over the generated data distribution $p_{\mathbf{g}}(\cdot)$ in the hope of cheating the discriminator that the image is indeed authentic. As the next step, these generated fake images and the real images sampled from the real data set are sent to the discriminator with proper labels, and the discriminator predicts $\mathrm{D}(x,\theta^D)$, probability of the image being real, where θ^D is the discriminator's parameters. The purpose of the discriminator is to distinguish fake and real ones and to give a feedback to the generator. The generator utilizes this feedback

*Correspondence: eneseken@aksaray.edu.tr

to decrease the "distance" between the generated data distribution $p_{\mathbf{g}}(\cdot)$ and real data distribution $p_{\mathbf{r}}(\cdot)$ to improve its capability of creating fake image that looks like authentic.

Depending on the definition of "distance" as the cost function, different variations of GANs have been proposed, e.g., Wasserstein GAN [3], maximum mean discrepancy GAN [4], Jensen–Shannon divergence which was used in vanilla GAN [2] and researchers have come up with different evaluation metrics, e.g., inception score (IS) [5], Fréchet inception distance (FID) [6] which uses inception network [7] to calculate Fréchet distance (FD) [8], average log-likelihood [2]. Among the many other evaluation metrics, particularly IS and FID are widely accepted by the researchers [9]. However, neither IS nor FID can catch the overfitting and underfitting in the GANs [10]. IS measures the Kullback–Leibler divergence between marginal class distribution and conditional class distribution, given the generated image. Theoretically, if the GAN generates only one good image from each class, that means the GAN is overfit, the IS will be very high [11], which is very misleading. On the other hand, FID measures the distance between generated data distribution and real data distribution which is a one-dimensional data. If a GAN's FID value is low (or high), both of the overfitting and uderfitting can be the reason of being low (or high) FID value.

In this paper, a quantitative method was proposed in order to determine overfitting and underfitting in the GANs using the observation that derivative of approximate FD is negative if the GAN is overfit. In this case the generated data distribution only partially covers the real data distribution which can be described as the GAN generates realistic-looking fake images but the diversity of the images is low. Similarly derivative of FD is positive if the GAN is underfitting, namely, the generated images are very diverse but unfortunately do not look authentic. Theoretically, when the $p_{\mathbf{g}}(\cdot)$ becomes equal to $p_{\mathbf{r}}(\cdot)$, which is the ultimate goal of a GAN for a right fit, the derivative will be zero. Eventhough reaching to zero might not be possible every time in practice, out of two models with the same sign, we can safely be in favor of the one closer to zero as it will also close to right fit. In order to evaluate many different GAN models proposed by the researchers in an objective manner, such a metric showing overfitting/underfitting plays a crucial role. Since this method does not depend on any particular network, e.g., inception network [7], and can be used for any neural network based discriminator, the FD will be referred to describe the method at the rest of the paper instead of FID.

2. Preliminaries

A good evaluation metric is critically important for GANs to guide the researchers for better GAN models. Considering the role played by ImageNet Large Scale Visual Recognition Challenge [12] (ILSVRC) at the advancement of CNNs for image classification models, the importance of an evaluation metric can be understood better. In this respect, many different evaluation metrics have been proposed so far by the researchers [9], including recovery error [13], IS [5], FID [6], and Kernel inception distance (KID) [14]. Although the IS and FID gained gradually increasing reputations and are commonly used, unfortunately both of them fail when it comes to determine whether the proposed GAN model is overfitting or underfitting.

In order to detect overfitting, in [13], using recovery error was proposed. By optimizing the random input vector $\boldsymbol{z}$, a GAN model tries to generate mimic of each image in the training set and validation set, and obtains recovery error distributions between training set-generated images (mimic of training set) and validation set-generated images (mimic of validation set). If the two recovery error distributions are different from each other, it is said that the GAN model is overfitting. In this setting, for each image in the training and validation set, an optimization problem should be solved for finding the optimum value of $\boldsymbol{z}$ to mimic that image, which

could be very time consuming. In addition, it is not possible quantitatively discriminating underfitting and overfitting.

The IS uses an Inception Network [7] pretrained on ImageNet [12]. The IS measures the expected Kullback–-Leibler divergence between generated images' marginal label distribution $p(y)$ and conditional label distribution given the images $p(y|\mathbf{x})$ which can be expressed as:

$$\text{IS(G)}=\exp(\mathbb{E}_{\mathbf{x}\sim p_g}[\text{KL}(p(y|\mathbf{x}) \parallel p(y))]), \tag{1}$$

both of the $p(y)$ and $p(y|\mathbf{x})$ can be obtained by applying inception network to the generated images. IS favors the GAN models which are more certain about the conditional probability of label given the image (low entropy), and less certain about the marginal probability of label (high entropy). However, since the IS does not take into account the real data distribution $p_{\mathbf{r}}(\cdot)$, it could not capture whether the GAN model is overfit or underfit.

In order to take into account the $p_{\mathbf{r}}(\cdot)$, FID [6] was proposed. The FD between two multivariate Gaussian distributions $\mathcal{N}(\mu_r, \Sigma_r)$ and $\mathcal{N}(\mu_g, \Sigma_g)$ is defined as [8]:

$$FD(\mu_r, \Sigma_r, \mu_g, \Sigma_g) = \|\mu_r - \mu_g\|_2^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}). \tag{2}$$

FID calculates FD between real and generated images' probability distributions. Here, the mean vectors μ and the covariance matrices Σ are obtained by embedding a set of generated and real images ($\mathbf{x}_{g,r}$) in a learned feature space. This can be given by a specific layer of inception network's embedding function f as:

$$\begin{aligned} \mu_{g,r} &= \frac{1}{N_{g,r}} \sum_{i=0}^{N_{g,r}} f(\mathbf{x}_{g,r}^{(i)}) \\ \Sigma_{g,r} &= \frac{1}{N_{g,r}-1} \sum_{i=0}^{N_{g,r}} (f(\mathbf{x}_{g,r}^{(i)}) - \mu_{g,r})(f(\mathbf{x}_{g,r}^{(i)}) - \mu_{g,r})^T, \end{aligned} \tag{3}$$

where $N_{g,r}$ is the number of generated and real images. Lower FID means a smaller distance and hence, out of two GAN models, the one with the lower score is preferable.

In [14], it is argued that FID is a bias estimator, and the authors introduced an unbiased alternative to FID, the Kernel inception distance (KID). However, due to its high variance to be reliable [15], it has not been widely adopted.

Although FID takes into account the real data set and its statistics and became a standard [13] with these properties for GAN evaluation metric, since the FID is one-dimensional real value, by just looking at FID value, we cannot evaluate whether the GAN model is overfit or underfit.

In order to address this issue, we utilized the observation that the derivative of FD indicates whether the GAN model is overfitting or underfitting. Furthermore, in contrast to [13], for calculating the FD, only forward propagation, through the trained discriminator of the GAN, is needed which makes this method computationally more feasible for determining overfitting and underfitting.

3. Determining overfitting and underfitting in GAN models using Fréchet distance

One of the main reasons of developing a new GAN model is to obtain a $p_{\mathbf{g}}(\cdot)$ which converges to ground truth distribution of real data $p_{\mathbf{r}}(\cdot)$ as close as possible, because only in this way, creating fake images looking

authentic will be possible [16].

In the case that $p_{\mathbf{g}}(\cdot)$ (red dots) only partially covers $p_{\mathbf{r}}(\cdot)$ (green circles), as can be seen in Figure 1, the GAN model still will create realistic looking images, however the images will not be diverse which means that several different input vectors $\boldsymbol{z}$ will be assigned to same output which is defined as overfitting. On the contrary, if the $p_{\mathbf{g}}(\cdot)$ (blue stars) covers the $p_{\mathbf{r}}(\cdot)$ but also assigns high probabilities to some points which are not actually covered by the $p_{\mathbf{r}}(\cdot)$, the generated images will be very diverse, however this time the images will not be looking as real images which is defined as underfitting.

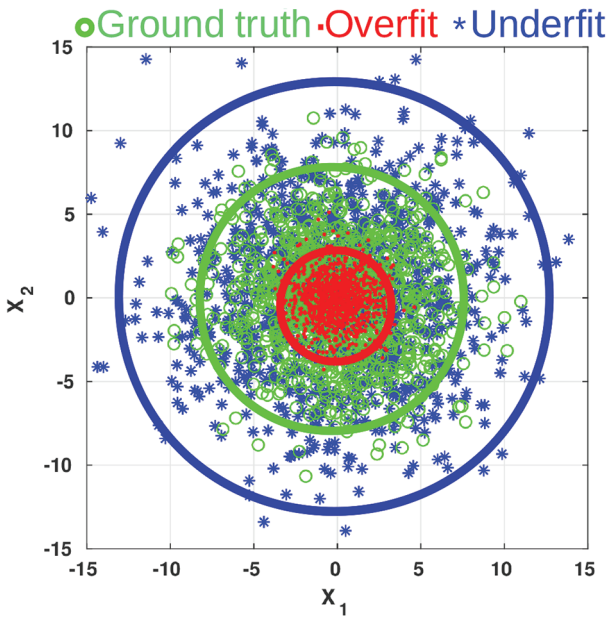

Figure 1. Demonstration of overfit and underfit distributions with respect to ground truth distribution. Here, the Fréchet distances of the overfit and underfit to the ground truth distribution are the same. Here the ground truth distribution represents the $p_{\mathbf{r}}(\cdot)$.

Although the Fréchet distance can be used to measure dissimilarity between two distributions, since it is symmetric, it assigns the same distance value, here for example 5.0, to both overfit ground truth and underfit ground truth distributions shown in Figure 1 (Here, the ground truth, overfit, and underfit distributions are assumed to have multivariate Gaussian distributions with $\mathcal{N}(\mathbf{0}, \Sigma = 11 * \mathrm{I})$, $\mathcal{N}(\mathbf{0}, \Sigma = 3.01 * \mathrm{I})$, and $\mathcal{N}(\mathbf{0}, \Sigma = 24 * \mathrm{I})$, with respectively.).

The FD between two distributions while keeping one of them the same (real data distribution, $p_{\mathbf{r}}(\cdot)$) and changing the other distribution's (generated data distribution, $p_{\mathbf{g}}(\cdot)$) covariance matrix (Σ) as a function of scalar θ can be reformulated as:

$$FD_{\theta}(\theta; \mu_r, \Sigma_r, \mu_g, \Sigma_g) = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g^{'}(\Sigma_g, \theta) - 2(\Sigma_r \Sigma_g^{'}(\Sigma_g, \theta))^{\frac{1}{2}}). \tag{4}$$

Here, $\Sigma_g^{'}(\Sigma_g, \theta)$ is defined as

$$\Sigma_g^{'}(\Sigma_g, \theta) = (\Sigma_g + \theta * \mathbf{I}) = (\sigma_g + \theta) * \mathbf{I} = \sigma_g^{'} * \mathbf{I}, \tag{5}$$

assuming that $\Sigma_r = \sigma_r * \mathbf{I}$, $\Sigma_g = \sigma_g * \mathbf{I}$, $\sigma_r > \sigma_g$, $\mu_r = \mu_g$ and $\theta, \sigma_r, \sigma_g \in \mathbb{R}^+$. The FD_θ between these two multivariate Gaussian distributions as a function of θ can be seen in Figure 2.

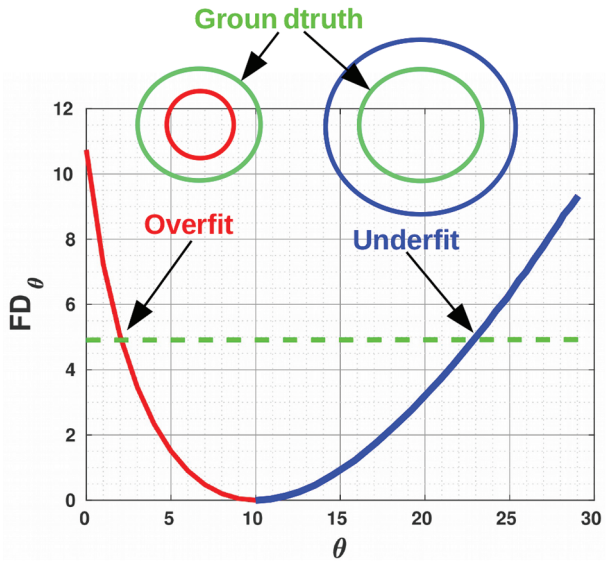

Figure 2. Demonstration of Fréchet distance between two given multivariate Gaussian distributions as a function of θ as explained at equation 4. Here, the ground truth distribution is assumed to have $\sigma_r = 11$ and the other distribution's σ_g value is 1 and its covariance matrix changes as a function of θ.

Roughly speaking, starting from a relatively small σ_g (here, $\sigma_g = 1$) and increasing the value of θ gradually, the distance between these two distributions will decrease with a negative slope until reaching a global minimum point where $\Sigma_g^{'}$ becomes equal to Σ_r. Theoretically, this global minimum point where the slope is zero, is an indicator of the right fit of a GAN model.

The FD value will be zero at this point if the $\mu_r = \mu_g$ and a nonnegative value otherwise. After this point if we keep going on increasing the θ, $\sigma_g^{'}$ will be greater than σ_r and the Fréchet distance will start to increase again with a positive slope.

Even though the Fréchet distance is not capable of telling right fit, overfit and underfit solely, the slope of the FD_θ can be utilized for this purpose. If we take the derivative of FD_θ with respect to θ, we can reach that

$$\begin{aligned} \frac{\partial FD_\theta}{\partial \theta} &= \frac{\partial(\mathrm{Tr}(\theta\mathbf{I}) - 2\mathrm{Tr}(\Sigma_r\Sigma_g + \theta\Sigma_r)^{\frac{1}{2}})}{\partial\theta} \\ &= d - \frac{d\sigma_r}{\sqrt{\sigma_r\sigma_g + \theta\sigma_r}}, \end{aligned} \tag{6}$$

where Σ ,$\mathbf{I} \in \mathbb{R}^{dxd}$ and here, the linear property of Trace operation and the property that $\partial(\mathrm{Tr}(\mathbf{X})) = \mathrm{Tr}(\partial(\mathbf{X}))$ were used.

The graph of the partial derivative of FD_θ with respect to θ can be seen in Figure 3. While being negative of partial derivative of FD_θ with respect to θ is an indicator of overfitting, conversely being positive is an indicator of underfitting. Theoretically, the derivative should be zero for the right fit. The absolute value

of the derivative numerically determines the level of underfit or overfit.

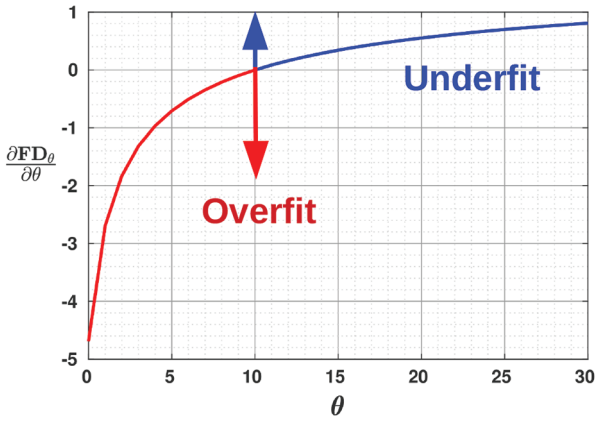

Figure 3. Partial derivative of FD_θ with respect to θ. In the overfit region the derivative will be negative, conversely in the underfit region the value will be positive.

Since FD_θ is a a convex function, value of θ that makes the FD_θ minimum will be

$$\theta^* = \arg\min_\theta \mathrm{FD}_\theta = \sigma_r - \sigma_g, \tag{7}$$

which makes the $\frac{\partial FD_\theta}{\partial \theta} = 0$. At this point the two distributions will be equal to each other if the mean vectors are also the same. In the case of the mean vectors not being the same, only the norm 2 of the difference of the mean vectors will be added to the FD_θ, as can be seen in Eq. 4, first term on the right hand side. This will result in an upward shift of the graph, but since this term is independent of θ, the curve and the derivative will be the same.

In practice, the covariance matrices Σ_r and Σ_g will not be diagonal matrices that might not be simplified as Eq. 6, however, the approximate derivative can still be calculated as

$$\frac{\partial FD_\theta}{\partial \theta} \approx \lim_{\theta \to 0} \frac{FD_\theta(0; \mu_r, \Sigma_r, \mu_g, \Sigma_g) - FD_\theta(\theta; \mu_r, \Sigma_r, \mu_g, \Sigma_g)}{\theta} \tag{8}$$

and the fact that

$$model = \begin{cases} \text{overfit, if} & \frac{\partial FD_\theta}{\partial \theta} < 0, \\ \text{right fit, if} & \frac{\partial FD_\theta}{\partial \theta} = 0, \\ \text{underfit, if} & \frac{\partial FD_\theta}{\partial \theta} > 0, \end{cases} \tag{9}$$

still holds true.

As can be seen from Figure 3, $\frac{\partial FD_\theta}{\partial \theta}$ shows a logarithmic behaviour, namely, while the GAN model is overfit, a small change in θ axis results in a huge difference. Contrary, if the model shows an underfit behaviour, a large change in θ axis can only make a small difference in $\frac{\partial FD_\theta}{\partial \theta}$. Clearly, this kind of characteristic may lead a misunderstanding at the comparison of models. In order to prevent this, we can simply take the exp of $\frac{\partial FD_\theta}{\partial \theta}$

which results in a linear scale as can be seen in Figure 4 and can be restated as

$$model = \begin{cases} \text{overfit, if} & \exp\left(\frac{\partial FD_\theta}{\partial \theta}\right) < 1, \\ \text{right fit, if} & \exp\left(\frac{\partial FD_\theta}{\partial \theta}\right) = 1, \\ \text{underfit, if} & \exp\left(\frac{\partial FD_\theta}{\partial \theta}\right) > 1. \end{cases} \tag{10}$$

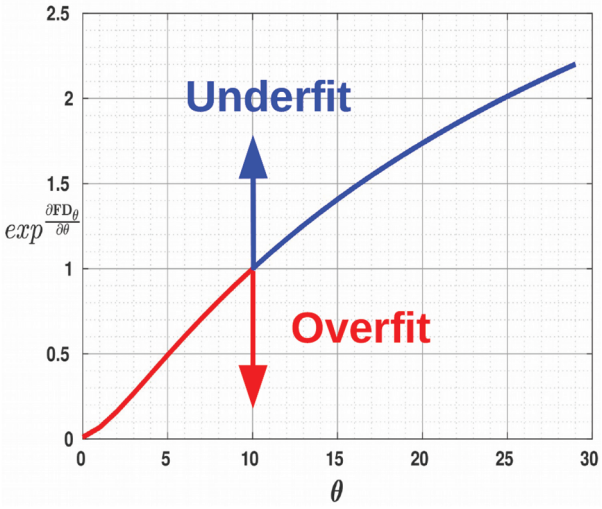

Figure 4. In order to prevent a misunderstanding at the comparison of GAN models based on $\frac{\partial FD_\theta}{\partial \theta}$ which has a logarithmic shape, by taking the exp of $\frac{\partial FD_\theta}{\partial \theta}$, we can project it to approximately linear scale.

4. Results and discussions

4.1. Implementation details of the GAN models

In the simulations, two popular GAN architectures, the conditional GAN (CGAN) [17] and deep convolutional GAN (DCGAN) [18] and two commonly preferred cost functions, Jensen–Shannon and Wasserstein-1 were used in order to demonstrate that the proposed method works very well independent of the design architecture and the cost functions. The CGAN is conditional version of DCGAN which greatly leverages the benefits of CNNs in both generative and discriminator part of the GAN. Different than the DCGAN, in CGAN, the generator and the discriminator networks are fed with the input vector $\boldsymbol{z}$ and the class conditional one-hot label vector. In this way, the generator network only produces images belong to a specific class, and the discriminator network utilizes this conditional vector to differentiate fake and real images in a specific class. In the DCGAN, however, generated images can be belonged to any classes.

In the following experiments, simulations were conducted for MNIST and CIFAR10 datasets. The design architecture for MNIST simulations is as follows: The dimension of the random input vector $\boldsymbol{z}$ of generator network was set to 100 and assumed that $\boldsymbol{z}$ follows a uniform distribution between $[-1, +1]$. Afterward, $\boldsymbol{z}$ was concatenated with the conditional one-hot label (for CGAN) vector before the dense layer with $7X7X128$ neurons which later reshaped as 128 filters with size of $7X7$. Following this step, 4 stack of Batch-Normalization

(BN) [19], Rectified linear unit (ReLU) [20], and transposed 2-D convolutional layer were used to generate fake images. The number of filters at each stack are set as 128, 64, 32, 1, with kernel size of 5. As the activation function the sigmoid was used since in experiences it converges faster than tanh although tanh was proposed in [18].

On the discriminator network, the conditional one-hot label vector (for CGAN), connected to $28X28X1$ number of dense neurons before reshaped as 1 filter with size of $28X28X1$ and concatenated with the input image with same size. Afterward, 4 stack of (ReLU) [20] and convolutional layers with 32, 64, 128, and 256 filters were used with kernel size of 5, and flattened before the output layer with one neuron.

The GAN design architecture used for CIFAR10 simulations, different than the used for MNIST simulations, is as follows: The random input vector $\boldsymbol{z}$ was concatenated with the conditional one-hot label (for CGAN) vector before the dense layer with $2X2X512$ neurons which later reshaped as 512 filters with size of $2X2$. Following this, 4 stack of transposed 2-D convolutional layers were used with $256, 128, 64, 3$ filters and the size of the filters were doubled at each stack as $2X2, 4X4, 8X8, 16X16$, and $32X32$. As activation function Leaky ReLU [20] was used. The kernel size of the filters was set as 5. For the discriminator network, the conditional one-hot label vector connected to $32X32X3$ dense neurons before reshaped as 3 filters with size of $32X32$ and concatenated with the input image with size of $32X32X3$. Afterward, 4 stack of 2-D convolutional layers were used with $64, 128, 256, 512$ filters with sizes of $16X16, 8X8, 4X4$, and $2X2$. The last layer was flattened before the output neuron with sigmoid function.

4.2. Data sets

As the training set, two commonly preferred data sets in the generative models, MNIST and CIFAR-10 [21], were used. The MNIST data set contains $60,000$ labeled gray-scale images of hand-written digits with size of $28X28X1$ pixels. CIFAR-10 data set contains $50,000$ colour images with 32323 size in 10 classes (cat, dog, bird, deer, frog, horse, ship, truck, airplane, automobile).

4.3. Results

In order to demonstrate the usefulness of FD in determining underfitting and overfitting in GANs, the three possible stages of training the GANs were depicted, namely, right fitting, overfitting and underfitting models. For comparison purpose, KID and IS of each model is also presented. In these simulations, CGAN was preferred, and as the cost function Jensen-Shannon was chosen. Simulations were conducted based on MNIST and CIFAR-10 datasets.

1. The GAN is trained just as right. In this desirable scenario, the generated data distribution closely replicates the real data distribution. The generated images will be realistic and diverse as can be seen in Figure 5. For a right fit training there are two indicators; First, the FD will be relatively small, second the absolute values of the derivatives of FD with respect to θ will be close to zero, as can be seen in Table 1 (compared to other GAN models which are underfit and overfit, for example as presented at Table 2 and Table 3, respectively). As depicted at Eq. 9, the $\frac{\partial FD_\theta}{\partial \theta}$ should be equal to zero for right fit, theoretically. Although reaching to zero might not be possible every time in practice, while comparing the two models, quantitatively we can favor in the one which is closer to zero. At this point, since the absolute values will be close to zero, the signs of the derivatives of different classes can be different, although it is not necessary, because of the randomness coming from the nature of the GANs. For example digit 2 in Table 1

has a negative value. However, even if it were positive, since the absolute value is small, we can still favor in this model compared to underfitting model which creates the results depicted at Table 2.

2. The GAN is underfitting. The model does not show its full capacity yet, possibly due to insufficient number of training epochs. In this case, even though the generated images will be diverse, they will also look unrealistic. Correspondingly the FDs will be relatively high, and the derivatives for all of the classes will be positive. An example of this scenario can be seen in Figure 6 and the correspondingly the FDs and the derivatives presented at Table 2 indicate this undefitting.

3. The GAN is overfitting. In this scenario, the generative model creates genuine images, but the randomly created images look identical in a given class.

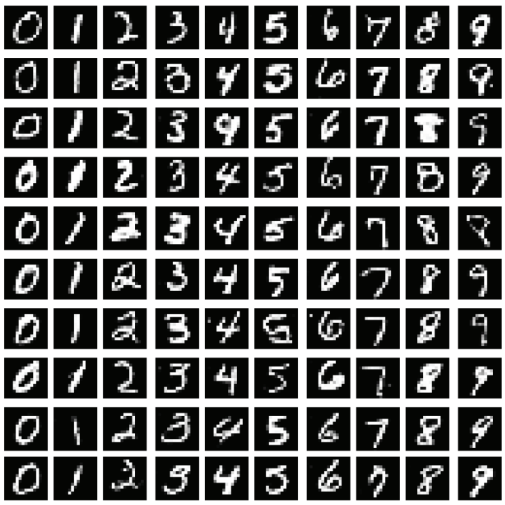

Figure 5. Demonstration of right fitting of a generative adversarial network. In this scenario the generated images look like genuine. In addition, in a specific class, the randomly generated images look like diverse.

Table 1. FDs and the derivatives for right fit.

	0	1	2	3	4	5	6	7	8	9
FD	0.80	0.42	1.81	0.99	1.08	2.02	1.19	1.11	2.04	0.68
$\frac{\partial FD_\theta}{\partial \theta}$	0.03	0.11	−0.63	0.20	0.59	0.29	0.54	1.27	0.79	0.76

Table 2. FDs and the derivatives for underfitting.

	0	1	2	3	4	5	6	7	8	9
FD	23.4	29.6	21.8	23.9	23.2	23.8	25.1	31.8	24.8	26.1
$\frac{\partial FD_\theta}{\partial \theta}$	2.72	2.91	2.85	2.67	2.74	2.77	2.92	3.20	2.65	2.87

The lack of diversity is a curical problem for generative models. Figure 7 shows an example of overfitting. In order to force the GAN model to overfit in this simulation, we simply restrict the size of training set. As can be seen from Table 3 the derivatives are all negative, as expected. In this simulation the overfitting

can be more clearly seen at digit 7, for example. Comparing with the digit $7s$ presented at Figures 5 and 7, right fitting and overfitting can be understood better.

Table 3. FDs and the derivatives for overfitting.

	0	1	2	3	4	5	6	7	8	9
FD	58.0	34.8	49.0	47.5	48.0	56.1	44.7	50.2	34.7	46.7
$\frac{\partial FD_\theta}{\partial \theta}$	-10.5	-2.7	-6.9	-6.0	-9.29	-11.0	-5.09	-5.75	-2.70	-6.96

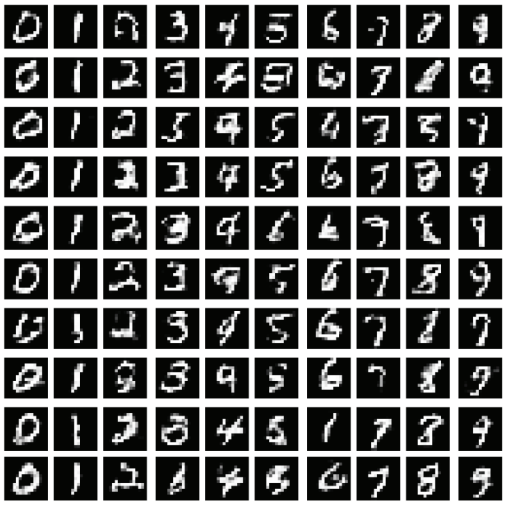

Figure 6. If the GAN model is underfitting, the created images will show artificial effects, even if they come from a diverse range. In this case the FDs will be high, and $\frac{\partial FD_\theta}{\partial \theta}$s for each class will be positive.

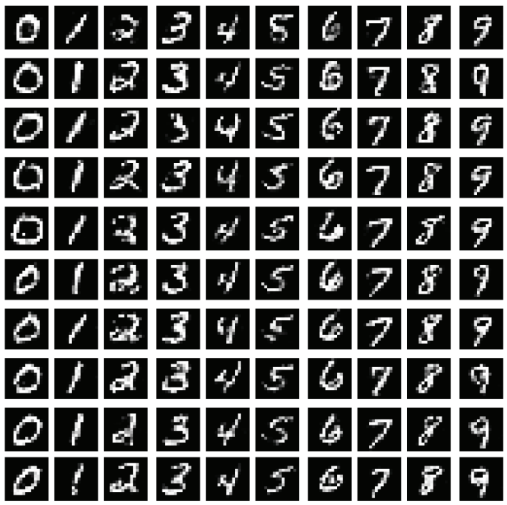

Figure 7. When the GAN model is overfitting, the generated images look very nice, however, at the same time, look very identical. In this case $\frac{\partial FD_\theta}{\partial \theta}$ will be negative as an indicator of overfitting.

Besides the FD values, the IS and KID values can be seen in Table 4. Here, the models evaluated based on the overall classes, but not in a specific class. Since the IS does not consider the training data distribution, it cannot catch the overfit, and assigns close values to overfit and right fit models.

In addition to MNIST dataset, we also conduct the similar simulations for CIFAR10 [21] using the CGAN model for three different training stages. Horse images generated by underfit and overfit models are depicted at Figure 8. Although the images generated by the underfit model are unrecognizable with many artificial effects, and the images created by the overfit model can be easily classified as horse, FDs of these two models are very close to each other, namely 57.5 and 61.2, respectively (in fact, FD score of underfit model is slightly better than the overfit model).

Table 4. IS and KID values of CGAN models for MNIST and CIFAR10 datasets.

IS / KID	Right fit	Under fit	Over fit
MNIST	5.7 / 0.45	4.5 / 1.1	5.6 / 0.49
CIFAR10	6.1 / 0.05	5.2 / 0.21	6.1 / 0.06

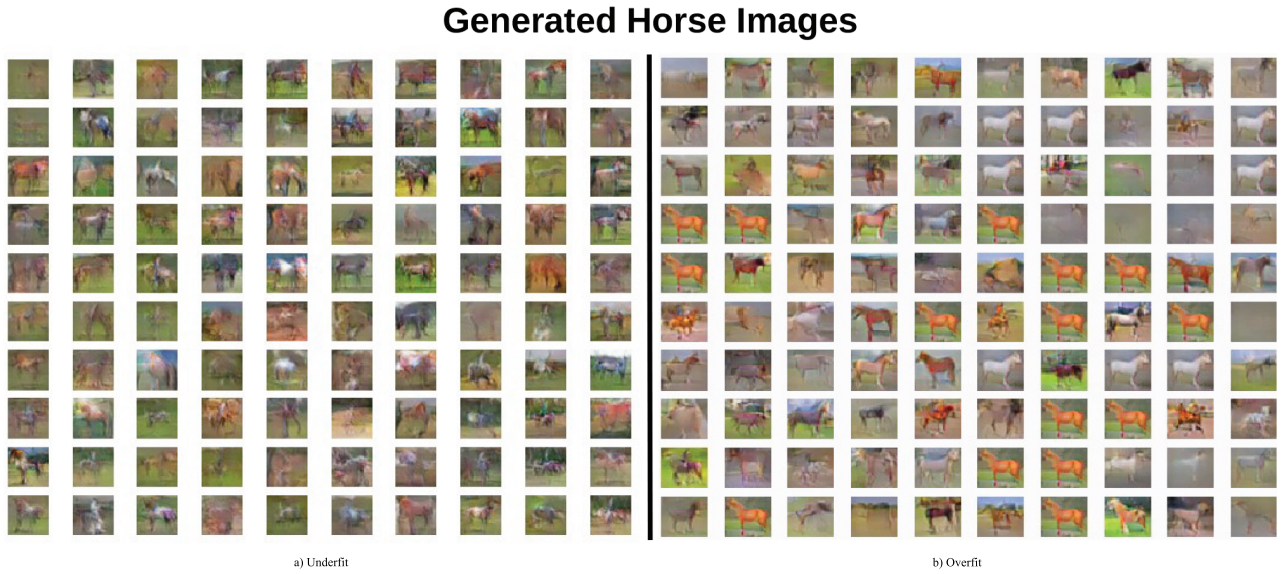

Figure 8. Generated horse images using the CGAN trained for CIFAR10. a) Depicts the images belong to underfit stage. These images are almost unrecognizable. b) Depicts the images belong to overfit stage. These images can be easily classified as horse. However, it should be noted that there are many similar images, for example, golden colour and white colour horses are dominant. Assigning different inputs to same output is a result of overfitting.

With this respect, in order to validate the two models in a quantitative way, just looking at the FD values could be misleading. However, the partial derivatives of the FD_θ with respect to θ clearly determines which which model is underfit and which one is overfit. For these simulations, for example, the $\frac{\partial FD_\theta}{\partial \theta}$s of underfit and overfit models are 15.7 and -12.5, respectively.

On the other hand, images generated by a CGAN model close to rightfit can be seen in Figure 9. As can be seen from the figure the generated horse images are very diverse in addition to being realistic looking. As supporting these results, the FD is 41.9 and $\frac{\partial FD_\theta}{\partial \theta}$ is 2.51. Although the $\frac{\partial FD_\theta}{\partial \theta}$ is not exatly zero, we can still

quantitatively evaluate that this model is closer to the right fit compared to the other ones. The related IS and KID values for CIFAR10 dataset can be seen in Table 4.

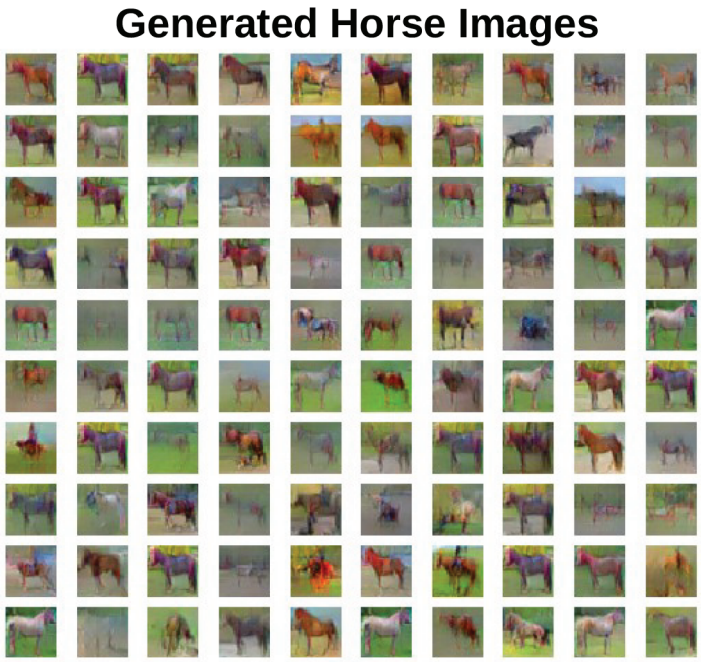

Figure 9. Generated horse images using the CGAN trained for CIFAR10, close to right fit.

To be able to employ this method, we only need the embedded distributions of the real and fake images and these distributions can be easily obtained from the dense layers of the discriminator network of a GAN model. For this reason conceptually, this method is independent of the GAN design (conditional or unconditional GANs) and the chosen cost function (Jensen–Shannon divergence, earth mover distance or Wasserstein-1, least square loss).

At the conducted simulations presented so far, CGANs were used, namely, it was possible to force the GAN to generated images belong to a specific class. However, since this method is independent of the condition, similar simulations can also be conducted with using the DCGAN [18]. For example Figure 10 shows the generated digits at the three different stages of DCGAN training. While the FD of the right fit model is 8.7, the underfit and over fit models' FDs are 35.8 and 23.4, respectively. On the other hand the $\frac{\partial FD_\theta}{\partial \theta}$s of underfit, right fit and overfit models are 23.5, 1.98, and -17.4, respectively. Again, by checking the signs, we are able to determine the overfit and underfit, and by looking the absolute values of FDs, we can tell which model is closer to the right fit training. That is being said, the conditional GAN models should be compared with the conditional ones and the unconditional GAN models should be compared with the unconditional counterparts for a fair evaluation [22].

In order to show the behavior of this method at the cost functions other than the Jensen–Shannon cost function, WGAN [3] can be used which utilizes the Wasserstein-1 cost function. Figure 11 shows the generated digits during the three different training stages of WGAN. The FDs of underfit, right fit and, over fit models are; 27.7, 10.3, and 15.4, respectively. The $\frac{\partial FD_\theta}{\partial \theta}$s on the other hand, are 18.9, -2.3, and -14.3, respectively, which indicates that the proposed method can be used safely with different cost functions. On the other hand,

for the comparison purposes, IS and KID values for these simulations are provided at Table 5.

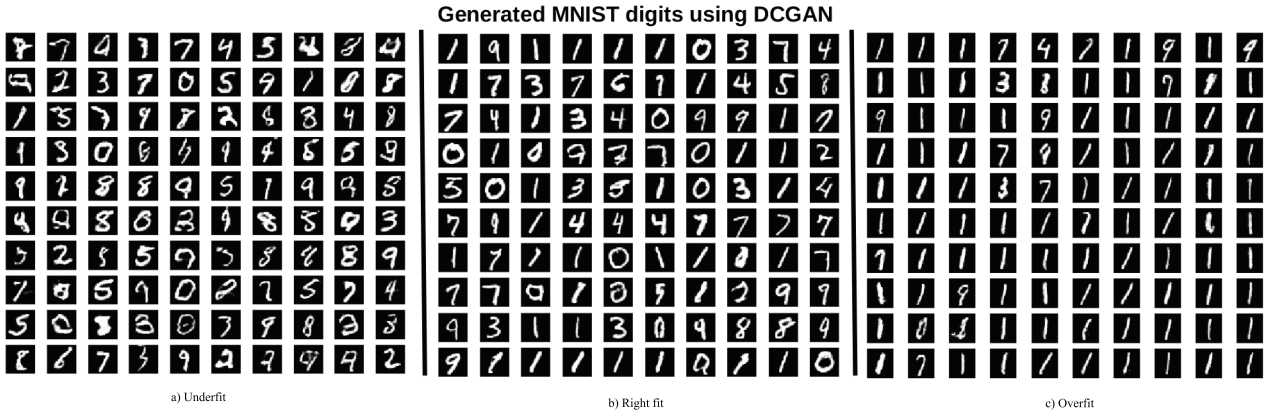

Figure 10. The proposed method is independent of the design architecture of the GAN. For this simualtions, DCGAN was used and a) underfit, b) right fit, and c) overfit stages were correctly determined by the proposed method.

Table 5. IS and KID values of DCGAN for MNIST dataset for different cost functions.

IS / KID	Right fit	Under fit	Over fit
JS	5.4 / 0.61	4.0 / 1.2	5.3 / 0.63
WGAN	5.5 / 0.57	4.1 / 1.0	5.5 / 0.55

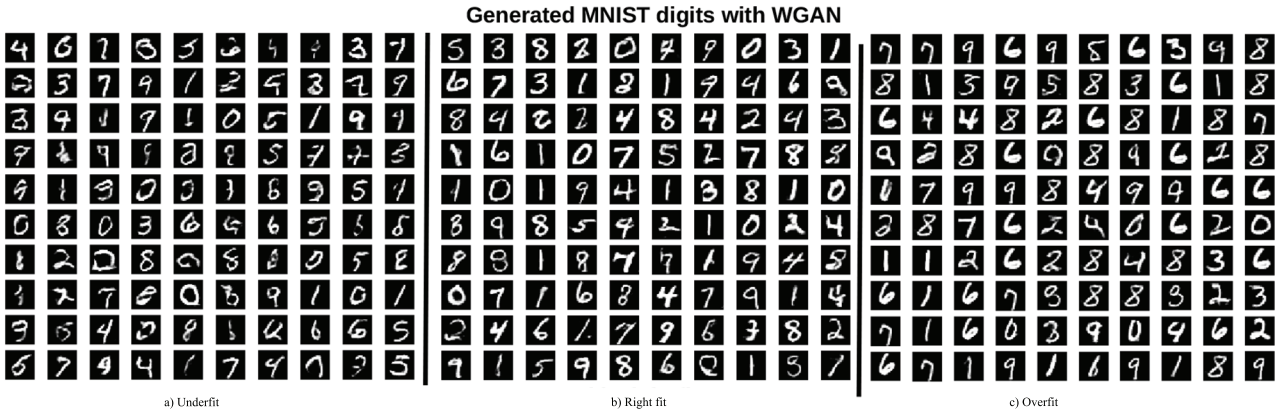

Figure 11. Generated images with Wasserstein-1 loss function. Images belong to overfit stage is more diverse compared to Figure 10c, however, digits $6s$ and $8s$ are almost identical.

5. Conclusion

With the rapid rise of GANs in many areas, researchers come up with many different GAN models. As a natural consequence of this, the question of "how to objectively evaluate different GAN models" has just emerged. In order to address this issue, Fréchet inception distance was proposed. Although FID is being commonly used by GAN research community, it fails to determine overfit and underfit. In this paper we demonstrated how

to approximately use FD_θ to detect whether a GAN model is overfit or underfit. Since FD_θ has a convex shape, the sign of $\frac{\partial FD_\theta}{\partial \theta}$ gives a direct information whether the model is overfit or underfit. With this tool, in addition to evaluate two models, researchers will also be able to quantitatively determine whether the model will generate diverse and authentic images which is the ultimate goal of a GAN model. Finally, we also showed the effectiveness of this method on MNIST and CIFAR-10 datasets, and we validated our method against to different cost functions such as Jensen–Shannon and Wasserstein-1, and different GAN architectures such as CGAN and DCGAN.

References

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

[2] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D et al. Generative adversarial nets. Advances in Neural Information Processing Systems 2014; 2672-2680.

[3] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN. arXiv 2017; arXiv:1701.07875v3.

[4] Li CL, Chang WC, Cheng Y, Yang Y, Póczos B. Mmd gan: towards deeper understanding of moment matching network. Advances in Neural Information Processing Systems 2017; 2203-2213.

[5] Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A et al. Improved techniques for training gans. Advances in Neural Information Processing Systems 2016; 2234-2242.

[6] Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in Neural Information Processing Systems 2017; 6626-6637.

[7] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016; 2818-2826.

[8] Dowson DC, Landau BV. The Fréchet distance between multivariate normal distributions. Journal of Multivariate Analysis 1982; 12 (3): 450-455.

[9] Borji A. Pros and cons of gan evaluation measures. Computer Vision and Image Understanding 2019; 179: 41-65.

[10] DeVries T, Romero A, Pineda L, Taylor GW, Drozdzal M. On the evaluation of conditional gans. arXiv preprint 2019; arXiv:1907.08175.

[11] Barratt S, Sharma R. A note on the inception score. In: ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models; Stockholm, Sweden; 2018.

[12] Deng J, Dong W, Socher R, Li LJ, Li K et al. Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition; Miami, FL, USA; 2009. pp. 248-255.

[13] Webster R, Rabin J, Simon L, Jurie F. Detecting overfitting of deep generative networks via latent recovery. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2019; 11273-11282.

[14] Bińkowski M, Sutherland DJ, Michael Arbel, Gretton A. Demystifying MMD GANs. In: International Conference on Learning Representations; Vancouver, BC, Canada; 2018. pp. 1-36.

[15] Ravuri S, Vinyals O. Seeing is not necessarily believing: limitations of biggans for data augmentation. In: ICLR Workshop on International Conference on Learning Representations; New Orleans, LA, USA; 2019.

[16] Theis L, Oord A, Bethge M. A note on the evaluation of generative models. In: International Conference on Learning Representations (ICLR 2016); San Juan, Puerto Rico; 2016. 1-10.

[17] Mirza M, Osindero S. Conditional generative adversarial nets. arXiv preprint 2014; arXiv:1411.178.

[18] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint 2015; arXiv:1511.06434.

[19] Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning; Lille, France; 2015. pp. 448-456.

[20] Nair V, Hinton GE. Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th International Conference on Machine Learning (ICML-10) 2010; 807-814.

[21] Krizhevsky A, Hinton G. Learning Multiple Layers of Features from Tiny Images (Tech. Report). Princeton, NJ, USA: Citeseer, 2009.

[22] Goodfellow I. NIPS 2016 tutorial: generative adversarial networks. arXiv preprint 2016; arXiv:1701.00160.