Statistical Distance Latent Regulation Loss for Latent Code Recovery

Jeongik Cho[1]

jeongik.jo.01@gmail.com[1]

Abstract

Finding a latent code that can generate specific data by inverting a generative model is called latent code recovery (or latent vector recovery). When performing gradient descent based latent recovery, the probability that the recovered latent code was sampled from a latent random variable can be very low. To prevent this, latent regulation losses or element resampling methods have been used in some papers.

In this paper, when the latent random variable is an IID (Independent and Identically Distributed) random variable and performing gradient descent-based latent code recovery, we propose statistical distance latent regulation loss to maximize the probability that the latent code was sampled from the latent random variable. The statistical distance latent regulation loss is the distance between the discrete uniform distribution, assuming each latent code element has the same probability and one-dimensional distribution that each element of the latent random variable follows in common. Since the statistical distance latent regulation loss considers all elements simultaneously, it maximizes the probability that the latent code was sampled from a latent random variable.

Also, we propose the latent distribution goodness of fit test, an additional test that verifies whether the latent code is sampled from the latent random variable. This additional test verifies whether all recovered latent codes' elements' distribution follows one-dimensional distribution that each element of the latent random variable follows in common when the latent random variable is an IID random variable. Passing the latent distribution goodness of fit test does not mean that the latent codes are recovered correctly, but when the latent codes are recovered correctly, the latent distribution goodness of fit test should be passed.

Compared with other latent regulation losses or element resampling methods, only latent code recovery using the statistical distance latent regulation loss could recover the correct latent code with high performance in the gradient descent-based latent code recovery.

1. Introduction

Generator G of a generative model like GAN maps latent random variable Z to data random variable X. In general, latent random variable Z follows a simple probability distribution such as $U(a,b)^{d_z}$ or $N(\mu,\sigma^2)^{d_z}$, and data random variable X follows the probability distribution of the complex data to be trained. d_z is the dimension of a latent random variable. Finding the ideal latent code z^* that can generate a data x which is sampled from the data random variable X by inverting the trained generator G is called latent code recovery (or latent vector recovery).

In this paper, when the latent random variable Z of generator G is an IID random variable and performing gradient descent-based latent code recovery, we propose the statistical distance latent regulation loss to maximize the probability that the latent code z_p was sampled from the latent random variable Z. The statistical distance latent regulation loss is the distance between the discrete uniform distribution, assuming that each element of the latent code z_p has the same probability and one-dimensional distribution that each element of the latent random variable Z follows in common. Since the statistical distance latent regulation loss considers all elements simultaneously, it maximizes the probability that the latent code z_p was sampled from a latent random variable Z.

Also, we propose the latent distribution goodness of fit test, an additional test that verifies whether the latent code z_p is sampled from the latent random variable Z. This additional test verifies whether all recovered latent codes' elements' distribution follows one-dimensional distribution that each element of the latent random variable Z follows in common when the latent random variable Z is an IID random variable. Passing the latent distribution goodness of fit test does not mean that the latent codes are recovered correctly, but when the latent codes are recovered correctly, the latent distribution goodness of fit test should be passed.

Compared with other latent regulation losses or element resampling methods, only latent code recovery using the statistical distance latent regulation loss could recover the correct latent code with high performance in the gradient descent-based latent code recovery.

2. Related Works

There are gradient descent-based methods [1, 2, 3, 4, 5], encoder-based methods [6, 7, 8, 9, 10], and hybrid methods [19, 20] for latent code recovery.

The encoder-based method requires additional encoder training for latent code recovery. Also, when latent code recovery is performed on unseen data [24], adversarial attacked data [21], or abnormal data [22], the encoder-based method may not show good performance. Instead, the encoder-based method is much faster in inference than the gradient descent-based method.

The gradient descent-based method is slower than the encoder-based method but is more robust to unseen data or out-of-distribution data such as adversarial attacked data or abnormal data.

The hybrid method roughly estimates the latent code z_p using an encoder, and then estimates the more accurate latent code z_p through gradient descent method.

Gradient descent-based latent code recovery is a method of repeatedly performing gradient descent on latent code z_p to reduce reconstruction loss L_{rec}, which is an error between generated data $G(z_p)$ and input data x. The following function shows the gradient descent-based latent code recovery [5] process.

$function\ latent\_code\_recovery(x, G, n, opt)$:

$z_p \leftarrow init(\,)$

$repeat\ n\ times$:

$L_{rec} \leftarrow diff\left(x, G(z_p)\right)$

$L \leftarrow L_{rec}$

$z_p \leftarrow z_p + opt\left(-\frac{\Delta L}{\Delta z_p}\right)$

$return\ z_p$

Fig.1 Gradient descent-based latent code recovery function

$init$ is a function that initializes the values of z_p. n is the number of times to perform gradient descent. opt is an optimizer that performs gradient descent. $diff$ is a function that measures the difference between two data. In [5], the performance when using different $diff$ functions was compared. L_{rec} is the reconstruction loss. L is the total loss. Through the above function, the latent code z_p that minimizes the reconstruction loss L_{rec} for the data x can be found.

However, it cannot be said that the latent code z_p found correctly represents data x, except for general problems in gradient descent optimization (e.g., local optimum problem). Since generator G is not trained to generate out-of-distribution data, there is always a tendency to generate data random variable X. Therefore, $P(Z|z_p)$ (the probability that the recovered latent code z_p was sampled from the latent random variable Z) can be very low. For example, the following figure is an image generated by the generator of GAN that trained MNIST handwriting data with latent random variable $Z \sim U(-1,1)^{256}$.

Fig.2 Generated data by GAN trained with latent random variable $Z \sim U(-1,1)^{256}$

The FID [11] between the test MNIST images and generated images through this GAN is 5.64053. FID indicates how close the generated data random variable $G(Z)$ and the data random variable X are. Therefore, if the generator G generates in-distribution data well, it will have a low FID.

When $Z'{\sim}U(-10,10)^{256}$ is input to this trained generator G, the following data is generated.

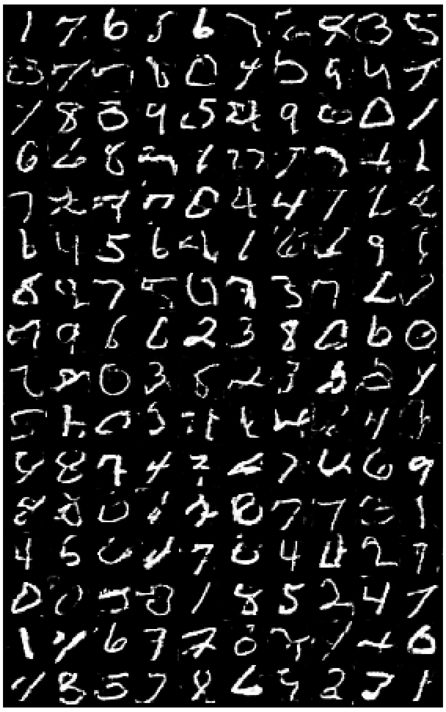

Fig.3 Generated data by GAN with input $Z'{\sim}U(-10,10)^{256}$

You can see that many data looks like in-distribution data. When the input is $Z'{\sim}U(-10,10)^{256}$, the FID between the generated MNIST image and the test MNIST image is 40.521896. That is, generator G of trained GAN tends to generate in-distribution data even for latent code z_k with low $P(Z|z_p)$. Therefore, $P(Z|z_p)$ can be very low for a latent code z_p that is a local optimum or a global optimum that sufficiently minimizes the reconstruction loss L_{rec}.

Latent code z_p with low $P(Z|z_p)$ cannot be considered to represent the data x correctly. Such latent code z_p may cause problems in performing data interpolation [23] or data edit [24] using latent code. Therefore, an additional term is needed to maximize $P(Z|z_p)$ when performing gradient descent-based latent code recovery.

To maximize $P(Z|z_p)$, [1, 2] added latent regulation loss to loss L, and [3, 4] resampled some elements of latent code z_p after each gradient descent.

The following function shows latent code recovery using latent regulation loss [1, 2].

$function\ latent\_recovery(x, G, n, opt)$:

$z_p \leftarrow init(\,)$

$repeat\ n\ times$:

$L_{rec} \leftarrow diff\left(x, G(z_p)\right)$

$L \leftarrow L_{rec} + \lambda_{lr} L_{lr}$

$z_p \leftarrow z_p + opt\left(-\frac{\Delta L}{\Delta z_p}\right)$

$return\ z_p$

Fig.4 Latent code recovery function with a latent regulation loss

L_{lr} is the latent regulation loss, and λ_{lr} is the latent regulation loss weight, respectively.

In the paper [1], z score square latent regulation loss was used when the latent random variable $Z{\sim}N(\mu,\sigma^2)^{d_z}$.

$$L_{lr} = \left(\frac{z_p - \mu}{\sigma}\right)^2$$

However, z score square latent regulation loss maximizes not $P(Z|z_p)$, but $\sum_{i=1}^{d_z} P(Z[i]|z_p[i])$. $z_p[i]$ is i-th element of z_p, and $Z[i]$ is i-th element of Z. Therefore, correct latent code recovery cannot be achieved with z score square latent regulation loss.

In the paper [2], the fool discriminator latent regulation loss was used.

$$L_{lr} = L_{adv}^{g}$$

L_{adv}^{g} is the adversarial loss of generator G. For example, if GAN was trained with the adversarial loss of LSGAN [13], then $L_{lr} = \left(D\left(G(z_p)\right) - 1\right)^2$.

D is a trained discriminator. However, fool discriminator latent regulation loss does not maximize $P(Z|z_p)$, and there is no guarantee that that the adversarial loss L_{adv}^{g} of generator G is minimized when $P(Z|z_p)$ is maximized. Moreover, since discriminator D is used to calculate the latent regulation loss L_{lr}, it requires much additional computation.

The following function shows latent code recovery using element resampling [3, 4].

$function\ latent\_recovery(x, G, n, opt)$:

    $z_p \leftarrow init(\,)$

    $repeat\ n\ times$:

        $L_{rec} \leftarrow diff\left(x, G(z_p)\right)$

        $L \leftarrow L_{rec}$

        $z_p \leftarrow z_p + opt\left(-\frac{\Delta L}{\Delta z_p}\right)$

        $z_p \leftarrow resampling(z_p)$

    $return\ z_p$

Fig.5 Latent code recovery function with element resampling

$resampling$ is a function that is resampling specific elements of latent code z_p from latent random variable Z. In the paper [3], when latent random variable $Z \sim U(a,b)^{d_z}$, boundary resampling was proposed in which all elements of latent code z_p out of range $[a,b]$ is resampled from $U(a,b)$ was proposed. However, boundary resampling maximizes $\sum_{i=1}^{d_z} P(Z[i]|z_p[i])$ like z score square, not $P(Z|z_p)$.

In the paper [4], when the latent random variable $Z \sim N(\mu,\sigma^2)^{d_z}$, stochastic resampling was proposed in which each element of the latent code z_p is stochastically resampled from $N(\mu,\sigma^2)$. Each element is resampled according to the probability function proposed in the paper and the value of $z_p[i]$. The closer $z_p[i]$ to μ, the lower the probability of resampling.

For stochastic resampling, the following two resampling probability functions are used.

$$f_{lc}(z_p) = \frac{1}{1 + e^{-a(|z_p| - b)}}$$

$$f_{tc}(z_p) = \begin{cases} \dfrac{e^{-\frac{a^2}{2}}}{e^{-\frac{z_p^2}{2}}} & if\ |z_p| < a \\ 1\ otherwise \end{cases}$$

f_{lc} is a logistic cutoff function, and f_{tc} is a truncated normal cutoff function. The output of each function is the probability of resampling. In the above equation, for convenience, $Z \sim N(0,1^2)^{d_z}$ is assumed. When $Z \sim N(\mu,\sigma^2)$, use $\frac{z_p-\mu}{\sigma}$ instead of z_p. Stochastic resampling maximizes $\sum_{i=1}^{d_z} P(Z[i]|z_p[i])$ but does not maximize $P(Z|z_p)$, because the resampling probability decreases as each element of the latent code z_p is closer to μ.

3. Statistical distance latent regulation loss

In this paper, to maximize $P(Z|z_p)$, we propose a statistical distance latent regulation loss, a latent regulation loss that can be used when the latent random variable Z is an IID random. The statistical distance latent regulation loss is the distance between the one-dimensional distribution A that each element of the latent random variable Z follows in common ($Z\sim A^{d_z}$) and the discrete uniform distribution S, which assumes that each element of the latent code z_p has the same probability. The statistical distance latent regulation loss is as follows.

$$P_S(x) = \begin{cases} \frac{1}{d_z} \; if \; x \in z_p \\ 0 \; otherwise \end{cases}$$

$$L_{lr} = Dist(P_A, P_S)$$

$Dist$ is a function that calculates the statistical distance between two distributions. P_A is the probability density function of distribution A. P_S is the probability mass function of the discrete uniform distribution created by the latent code z_p.

Unlike most existing methods that independently consider each element of latent code z_p, since statistical distance latent regulation loss considers all elements of latent code z_p simultaneously, $P(Z|z_p)$ can be maximized. Also, since statistical distance latent regulation loss can be used when the latent random variable Z is an IID random variable, it can be used when $Z\sim U(a,b)^{d_z}$, $Z\sim N(\mu,\sigma^2)^{d_z}$, or any distribution $Z\sim A^{d_z}$.

The following table.6 summarizes the conditions required for each latent regulation loss or resampling method.

	Z~ALL	Z~IID	Z~N	Z~U
Statistical distance		O	O	O
Z score square			O	
Fool discriminator	O	O	O	O
Boundary resampling				O
Stochastic resampling			O	

Table.6 Conditions of Z required for each latent regulation loss or resampling method

"Z~ALL" in the table above means that the latent random variable Z can be used regardless of the distribution and "Z~IID" means that it can be used when latent random variable Z is an IID random variable.

4. Latent distribution goodness of fit test

The latent code z_p with low reconstruction loss L_{rec} is not always the ideal latent code z^*. To verify whether the latent code z_p is sampled from the latent random variable Z, we propose latent distribution goodness of fit test.

Suppose that the latent random variable Z follows a specific distribution A^{d_z} and that latent code recovery is ideally performed for k data. At this time, the distribution of all elements of all recovered latent codes ($k \times d_z$ elements), will follow distribution A. Latent distribution goodness of fit test verifies whether these all elements follow the distribution A. If the distribution of all elements of all latent codes does not follow distribution A, the latent code cannot be said to have been recovered correctly.

However, passing the latent distribution goodness of fit test does not mean that the latent code has been recovered correctly. For example, if latent code recovery is performed using a very low learning rate, latent code z_p will have little difference from the initial value. And if z_p is initialized with the value sampled from the latent random variable Z, it can pass the latent goodness of fit test. Therefore, reconstruction loss L_{rec} is still essential for evaluation. Latent distribution goodness of fit test is an additional test whether latent code z_p minimizing reconstruction loss L_{rec} has been properly recovered.

Also, even if the latent random variable Z is not an IID random variable, the latent distribution goodness of fit test can be performed. Elements with the same index of the recovered latent code z_p should have been sampled from the same distribution.

Therefore, it is possible to perform latent distribution goodness of fit test for each element's distribution with the same index. In this case, the latent distribution goodness of fit test is performed d_z times.

5. Material and methods

We conducted experiments to compare gradient descent-based latent code recovery performance of latent regulation losses and element resampling methods. The MNIST handwriting dataset was used for the experiment. Each pixel value of data was normalized to $[-1,1]$. In the experiment, the process of training the model, performing latent code recovery, and evaluating the performance was repeated 3 times. All figures in section 6. "Experimental results and discussion" are averages of three experiments.

5.1 Model train

We used GAN as a generative model for the experiment. GAN refers to the structure of DCGAN [15], and the latent random variable dimension $d_z = 256$. GAN was trained using LSGAN's adversarial loss, $optimizer = Adam(learning\ rate = 10^{-5}), epoch = 200, batch\ size = 32$.

5.2 Latent code recovery

Wasserstein distance [16] and energy distance [17] were used as statistical distances for statistical distance latent regulation loss. In the statistical distance latent regulation loss, P_A was approximated by sampling enough samples (10000 samples) from A when $Z \sim A^{d_z}$. In the logistic cutoff of stochastic resampling, two hyperparameters are required. For a fair comparison, b, one of the two hyperparameters, is fixed at 2, which has the best performance in [4]. z_p initialize function $init()$ is $sampling(Z)$. Sixteen latent codes per data were initialized and optimized in parallel, and among them, the latent code z_p with the lowest loss L is selected. For $diff$, mean absolute error, which obtained the best result in [5], was used. The number of gradient descent $n = 200$ and optimizer $opt = Adam(beta\_1 = 0.9, beta\_2 = 0.999)$.

5.3 Performance evaluation

Since the gradient descent-based latent code recovery takes a very long time to perform, the performance was evaluated by performing latent code recovery on only 1000 data randomly selected out of 10,000 test data per experiment. Classification accuracy, latent distribution goodness of fit test, L1 loss, and L2 loss were used for evaluation. The classifier used in the evaluation was trained with $optimizer = Adam(learning\ rate = 10^{-5}), epoch = 50, batch\ size = 32$. Two-sided KS-test (Kolmogorov–Smirnov test) [18] was used as the latent distribution goodness of fit test. The null hypothesis H_0 is "all elements of all recovered latent code z_p were sampled from the latent distribution".

6. Experimental results and discussion

6.1 Latent random variable follows the normal distribution

This section shows the experimental results when latent random variable $Z \sim N(0, 1^2)^{256}$. GAN's FID is 6.68971, and classifier's accuracy is 99.283%. The following table shows the difference in performance according to the learning rate when performing gradient descent-based latent code recovery without any regulation term.

No regulation		Learning rate				
		0.00001	0.0001	0.001	0.01	0.1
Latent mean		-0.002	0.001	0.000	0.000	0.006
Latent variance		0.997	0.997	1.001	1.332	19.424
Goodness of fit test p-value1		2.7%	87.9%	65.5%	0.0%	0.0%
Goodness of fit test p-value2		16.5%	14.8%	92.2%	0.0%	0.0%
Goodness of fit test p-value3		72.9%	78.6%	18.7%	0.0%	0.0%
L1 loss		168.398	122.935	37.668	18.596	20.578
L2 loss		248.109	165.683	28.192	7.948	10.088
Classifier accuracy		39.1%	63.9%	96.4%	99.2%	98.2%

Table 7. Latent code recovery performance without regulation term

Latent mean and latent variance in Table 7 represents the mean and variance of latent codes, respectively. Since $Z \sim N(0, 1^2)^{256}$, if the latent code is recovered correctly, the latent mean should be close to 0, and the latent variance should be close to 1. The goodness of fit test p-value is the p-value when the latent distribution goodness of fit test is performed on the recovered latent codes. The other values in the table are the three experiments' average values, but since the average value of the p-value is not meaningful, the three experiments' results were shown separately. If the significance probability is 5%, the null hypothesis H_0 (all elements of all recovered latent code z_p were sampled from the latent distribution) is rejected when the p-value is less than 5% and the alternative hypothesis H_1 (all elements of all recovered latent code z_p were not sampled from the latent distribution) is rejected when the p-value is above 95%. In the table, when the significance probability is 5%, it is indicated in red when the null hypothesis H_0 is rejected, yellow when no hypothesis is rejected, and green when the alternative hypothesis H_1 is rejected. The L1 loss and L2 loss represent the L1 loss and L2 loss between the test data and the generated data by recovered latent code. Classification accuracy is the ratio at which the trained classifier correctly classifies the generated data $G(z_p)$.

In Table 8, it can be seen that when the learning rate is too high, the latent code distribution is significantly distorted, resulting in high variance and low goodness of fit test p-value. When the learning rate is less than 0.001, the latent code is not sufficiently searched, so the latent variance is close to 1, and the latent distribution goodness of fit p-value is relatively high. Therefore, to sufficiently search latent code when comparing the performance of latent regulation loss or element resamplings, we used the $learning\ rate = 0.01$.

The following Table 9-14 shows the latent code recovery performance of latent regulation loss or element resampling when $learning\ rate = 0.01$.

Wasserstein distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		0.001	0.000	0.000	0.000	0.000
Latent variance		1.314	1.179	0.999	0.995	0.995
Goodness of fit test p-value1		0.0%	0.0%	100.0%	58.8%	56.2%
Goodness of fit test p-value2		0.0%	0.0%	100.0%	58.8%	54.4%
Goodness of fit test p-value3		0.0%	0.0%	100.0%	58.9%	56.3%
L1 loss		18.594	18.554	19.726	28.229	69.082
L2 loss		7.982	7.853	8.531	16.336	74.007
Classifier accuracy		99.1%	98.8%	99.0%	98.5%	85.8%

Table 8. Latent code recovery performance with Wasserstein distance latent regulation loss

Energy distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		0.001	0.000	0.000	0.000	0.000
Latent variance		1.319	1.228	1.021	0.996	0.996
Goodness of fit test p-value1		0.0%	0.0%	63.8%	99.5%	89.3%
Goodness of fit test p-value2		0.0%	0.0%	62.8%	99.3%	91.6%
Goodness of fit test p-value3		0.0%	0.0%	65.5%	99.3%	91.9%
L1 loss		18.579	18.492	19.421	29.574	81.093
L2 loss		7.881	7.751	8.314	17.660	92.669
Classifier accuracy		99.0%	99.1%	99.3%	98.7%	82.2%

Table 9. Latent code recovery performance with Energy distance latent regulation loss

Z score square		Latent regulation loss weight						
		0.001	0.0032	0.0057	0.01	0.1	1	10
Latent mean		0.000	0.001	0.001	0.000	0.000	0.001	0.001
Latent variance		1.261	1.124	0.994	0.813	0.091	0.016	0.005
Goodness of fit test p-value1		0.0%	0.0%	18.3%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value2		0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value3		0.0%	0.0%	32.5%	0.0%	0.0%	0.0%	0.0%
L1 loss		18.584	18.413	18.425	18.204	17.171	22.391	51.088
L2 loss		7.862	7.737	7.751	7.480	6.128	10.173	46.626
Classifier accuracy		99.2%	99.1%	98.9%	98.9%	99.0%	99.1%	96.3%

Table 10. Latent code recovery performance with Z score square latent regulation loss

Fool discriminator		Latent regulation loss weight				
		0.000001	0.0001	0.01	1	100
Latent mean		-0.001	-0.001	-0.002	0.002	0.001
Latent variance		1.333	1.329	1.304	1.122	1.119
Goodness of fit test p-value1		0.0%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value2		0.0%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value3		0.0%	0.0%	0.0%	0.0%	0.0%
L1 loss		18.617	18.656	21.297	152.446	180.556
L2 loss		8.000	7.950	10.194	218.759	271.107
Classifier accuracy		99.1%	99.0%	98.8%	45.9%	38.4%

Table 11. Latent code recovery performance with fool discriminator latent regulation loss

Logistic cutoff		Hyperparameter				
		2	2.5	3	3.5	4
Latent mean		0.000	0.000	-0.001	-0.001	-0.001
Latent variance		0.330	0.281	0.261	0.265	0.282
Goodness of fit test p-value1		0.0%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value2		0.0%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value3		0.0%	0.0%	0.0%	0.0%	0.0%
L1 loss		99.463	73.065	57.442	46.068	39.747
L2 loss		119.189	74.780	50.483	34.795	26.771
Classifier accuracy		85.3%	93.0%	95.8%	97.6%	98.2%

Table 12. Latent code recovery performance with logistic cutoff element resampling

Truncated Normal cutoff		Hyperparameter				
		2	2.5	3	3.5	4
Latent mean		0.001	0.000	0.001	0.000	0.000
Latent variance		0.528	0.520	0.574	0.732	1.022
Goodness of fit test p-value1		0.0%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value2		0.0%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value3		0.0%	0.0%	0.0%	0.0%	0.0%
L1 loss		154.437	109.289	60.904	35.402	25.230
L2 loss		221.241	136.742	55.400	21.807	12.019
Classifier accuracy		50.4%	79.8%	95.3%	98.5%	99.0%

Table 13. Latent code recovery performance with truncated normal cutoff element resampling

When using appropriate latent regulation loss weight λ_{lr}, you can see only the statistical distance latent regulation loss (table 8-9) rejects the alternative hypothesis H_1 and has a low reconstruction loss L_{rec}. The z score square latent regulation loss weight was searched more closely than the statistical distance latent regulation loss weight. In z score square, it was possible to find a case where the null hypothesis H_0 could not be rejected, but the p-value was not high enough to reject the alternative hypothesis H_1. All other methods had low reconstruction loss L_{rec}, but did not pass the latent goodness of fit test.

6.2 Latent random variable follows the uniform distribution

The following tables show the performance according to the latent regulation loss when the latent random variable $Z \sim U(-1,1)^{d_z}$. The GAN's FID is 7.71122, and classifier's accuracy is 99.310%.

No regulation		Learning rate				
		0.00001	0.0001	0.001	0.01	0.1
Latent mean		0.000	0.001	0.001	-0.008	-0.119
Latent variance		0.333	0.333	0.341	0.571	19.105
Goodness of fit test p-value1		45.0%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value2		59.2%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value3		18.9%	0.0%	0.0%	0.0%	0.0%
L1 loss		165.593	101.444	30.046	18.102	23.047
L2 loss		243.748	128.611	19.032	7.321	12.479
Classifier accuracy		42.8%	74.7%	98.0%	99.2%	97.6%

Table 14. Latent code recovery performance without regulation term

Since $Z \sim U(-1,1)^{256}$, if the latent code is recovered correctly, the latent mean should be close to 0, and the latent variance should be close to $\frac{1}{3}$. As when $Z \sim N(0,1^2)^{256}$, it can be seen that latent variance is close to the ideal value, and the p-value is relatively high when the learning rate is low. The following Table 15-18 shows the latent code recovery performance of latent regulation loss or element resampling when $learning\ rate = 0.01$.

Wasserstein distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		-0.007	-0.003	0.000	0.000	0.000
Latent variance		0.564	0.500	0.346	0.333	0.333
Goodness of fit test p-value1		0.0%	0.0%	0.0%	99.9%	100.0%
Goodness of fit test p-value2		0.0%	0.0%	0.0%	99.9%	100.0%
Goodness of fit test p-value3		0.0%	0.0%	0.0%	100.0%	100.0%
L1 loss		18.282	18.194	18.357	22.330	40.748
L2 loss		7.530	7.442	7.297	10.489	31.589
Classifier accuracy		99.2%	98.8%	99.2%	98.9%	95.9%

Table 15. Latent code recovery performance with Wasserstein latent regulation loss

Energy distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		-0.007	-0.005	0.000	0.000	0.000
Latent variance		0.566	0.514	0.363	0.334	0.333
Goodness of fit test p-value1		0.0%	0.0%	0.0%	62.5%	100.0%
Goodness of fit test p-value2		0.0%	0.0%	0.0%	67.8%	100.0%
Goodness of fit test p-value3		0.0%	0.0%	0.0%	67.6%	100.0%
L1 loss		18.146	17.943	18.126	24.545	56.011
L2 loss		7.473	7.235	7.152	12.511	53.851
Classifier accuracy		98.8%	99.1%	99.4%	98.7%	91.5%

Table 16. Latent code recovery performance with energy latent regulation loss

Fool discriminator		Latent regulation loss weight				
		0.000001	0.0001	0.01	1	100
Latent mean		-0.007	-0.006	-0.008	-0.003	-0.003
Latent variance		0.572	0.573	0.557	0.450	0.451
Goodness of fit test p-value1		0.0%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value2		0.0%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value3		0.0%	0.0%	0.0%	0.0%	0.0%
L1 loss		18.266	18.155	20.489	160.519	189.300
L2 loss		7.473	7.435	9.367	233.935	287.610
Classifier accuracy		99.1%	99.0%	99.0%	44.7%	35.0%

Table 17. Latent code recovery performance with fool discriminator latent regulation loss

Boundary resampling		Learning rate				
		0.001	0.01	0.1	1	10
Latent mean		0.002	-0.003	-0.008	-0.005	0.000
Latent variance		0.268	0.247	0.297	0.320	0.333
Goodness of fit test p-value1		0.0%	0.0%	0.0%	0.0%	75.3%
Goodness of fit test p-value2		0.0%	0.0%	0.0%	0.0%	70.1%
Goodness of fit test p-value3		0.0%	0.0%	0.0%	0.0%	62.7%
L1 loss		38.924	22.434	47.876	92.122	176.282
L2 loss		29.258	10.167	37.701	108.229	263.273
Classifier accuracy		97.7%	99.1%	97.0%	83.9%	36.0%

Table 18. Latent code recovery performance with boundary resampling

In boundary resampling, since an additional hyperparameter is not required, we searched for a learning rate instead.

When using appropriate latent regulation loss weight λ_{lr}, as when $Z \sim N(0, 1^2)^{256}$, you can see only the statistical distance latent regulation loss (table 15-16) rejects the alternative hypothesis H_1 and has a low reconstruction loss L_{rec} . All other methods had low reconstruction loss L_{rec}, but did not pass the latent goodness of fit test. In boundary resampling, when the learning rate is high, the latent code element is almost always resampling, so it has a meaningful p-value, but L_{rec} is too high.

6.3 Latent random variable follows the unique distribution

In this section, to show that the statistical distance latent regulation loss can be applied even when the latent random variable Z is any IID random variable, we performed latent code recovery using statistical distance latent regulation loss and generator G trained with a unique IID random variable ($Z \sim A^{256}$). A is a half uniform and a half normal distribution. The probability density function of A is

$$P_A(x) = \begin{cases} 0 \; if \; x < -1 \\ 0.5 \; if \; x \in [-1,0] \\ \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \; if \; 0 < x \end{cases}.$$

The following graph shows the graph of the probability density function of A.

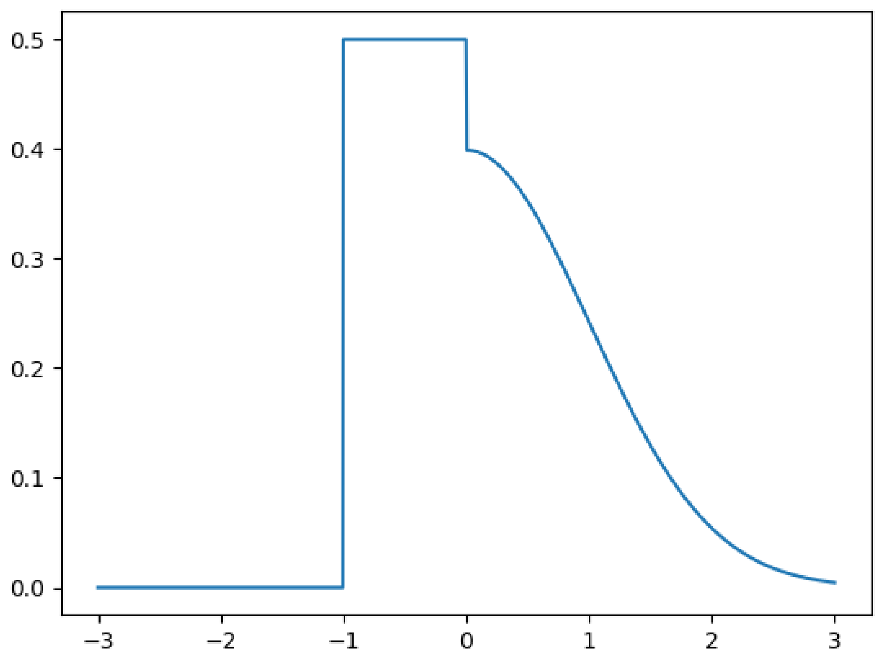

Figure 19. The probability density function of half uniform and half normal

The FID of GAN trained with this unique IID random variable is 6.70161, and the accuracy of the classifier is 99.317%.

No regulation		Learning rate				
		0.00001	0.0001	0.001	0.01	0.1
Latent mean		0.149	0.146	0.153	0.261	1.725
Latent variance		0.644	0.640	0.647	0.911	17.514
Goodness of fit test p-value1		6.5%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value2		48.7%	0.0%	0.0%	0.0%	0.0%
Goodness of fit test p-value3		23.4%	0.0%	0.0%	0.0%	0.0%
L1 loss		167.302	115.088	35.205	19.238	22.171
L2 loss		245.155	150.954	24.815	8.211	11.007
Classifier accuracy		39.7%	67.1%	97.9%	98.9%	98.8%

Table 20. Latent code recovery performance without regulation term

As in previous experiments, it was impossible to recover the latent code to have a low L_{rec} and a high p-value without a latent regulation loss.

The following tables 21-22 show the performance when using the statistical distance latent regulation loss when $learning\ rate = 0.01$, as in previous experiments.

Wasserstein distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		0.255	0.214	0.146	0.148	0.149
Latent variance		0.902	0.821	0.649	0.642	0.642
Goodness of fit test p-value1		0.0%	0.0%	0.0%	56.6%	56.9%
Goodness of fit test p-value2		0.0%	0.0%	0.0%	56.8%	55.9%
Goodness of fit test p-value3		0.0%	0.0%	0.0%	57.8%	56.1%
L1 loss		19.261	18.927	18.582	25.176	54.512
L2 loss		8.147	7.839	7.352	12.907	50.647
Classifier accuracy		99.1%	99.0%	99.0%	98.7%	92.4%

Table 21. Latent code recovery performance with Wasserstein distance latent regulation loss

Energy distance		Latent regulation loss weight				
		0.001	0.01	0.1	1	10
Latent mean		0.257	0.226	0.147	0.148	0.149
Latent variance		0.906	0.859	0.674	0.642	0.642
Goodness of fit test p-value1		0.0%	0.0%	0.0%	26.1%	85.6%
Goodness of fit test p-value2		0.0%	0.0%	0.0%	36.8%	86.2%
Goodness of fit test p-value3		0.0%	0.0%	0.0%	27.4%	86.5%
L1 loss		19.370	18.936	18.574	26.611	69.963
L2 loss		8.222	7.853	7.367	14.115	74.431
Classifier accuracy		99.1%	98.9%	99.3%	98.4%	86.9%

Table 22. Latent code recovery performance with energy distance latent regulation loss

When $Z \sim A$, the p-value was not high enough to reject the alternative hypothesis H_1 as $Z \sim U$ or $Z \sim N$. However, you can see that the statistical distance latent regulation loss has a sufficiently low reconstruction loss L_{rec}, with a p-value high enough that the null hypothesis H_0 is not rejected. These results show that the statistical distance latent regulation loss can also be used for latent code recovery of generator G trained with a unique IID random variable Z.

7. Conclusion

In this paper, when the latent random variable is an IID random variable and performing gradient descent-based latent code recovery, we proposed statistical distance latent regulation loss to maximize $P(Z|z_p)$. Unlike other methods that maximize $\sum_{i=1}^{d_z} P(Z[i]|z_p[i])$, since statistical distance latent regulation loss maximizes $P(Z|z_p)$, it enables correct latent code recovery.

In this paper, we also proposed the latent distribution goodness of fit test, an additional test used to evaluate the performance of latent code recovery. The latent distribution goodness of fit test evaluates whether latent code recovery has been performed correctly.

Compared with other latent regulation losses or element resampling methods, only latent code recovery using the statistical distance latent regulation loss could recover the correct latent code with high performance in the gradient descent-based latent code recovery.

8. References

[1] A. Creswell and A. A. Bharath, "Inverting the Generator of a Generative Adversarial Network," in IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 7, pp. 1967-1974, July 2019, doi: 10.1109/TNNLS.2018.2875194. [Online]. Available: https://ieeexplore.ieee.org/document/8520899

Z score square

[2] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing,

M. Hasegawa-Johnson and M. N. Do, "Semantic Image Inpainting with Deep Generative Models," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 6882-6890, doi: 10.1109/CVPR.2017.728. [Online]. Available: https://ieeexplore.ieee.org/document/8100211

Fool discriminator

[3] https://arxiv.org/abs/1702.04782v2

[4] https://arxiv.org/abs/1810.03764

[5] A. Patro, V. Makkapati and J. Mukhopadhyay, "Evaluation of Loss Functions for Estimation of Latent codes from GAN," 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), Aalborg, 2018, pp. 1-6, doi: 10.1109/MLSP.2018.8517097. [Online]. Available: https://ieeexplore.ieee.org/document/8517097

[6] https://arxiv.org/abs/2004.00049

[7] https://arxiv.org/abs/1605.09782

[8] https://arxiv.org/abs/1609.07093

[9] https://arxiv.org/abs/1606.00704

[10] https://arxiv.org/abs/1907.02544

[11] https://arxiv.org/abs/1706.08500

[12] R. Webster, J. Rabin, L. Simon and F. Jurie, "Detecting Overfitting of Deep Generative Networks via Latent Recovery," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 11265-11274, doi: 10.1109/CVPR.2019.01153. [Online]. Available: https://ieeexplore.ieee.org/document/8953411

[13] https://arxiv.org/abs/1611.04076

[14] Yann LeCun, Corinna Cortes, Christopher J.C. Burges

THE MNIST DATABASE of handwritten digits

http://yann.lecun.com/exdb/mnist/

[15] https://arxiv.org/abs/1511.06434

[16] https://arxiv.org/abs/1806.05500

[17] http://pages.stat.wisc.edu/~wahba/stat860public/pdf4/Energy/EnergyDistance10.1002-wics.1375.pdf

[18] (2008) Kolmogorov–Smirnov Test. In: The Concise Encyclopedia of Statistics. Springer, New York, NY. https://doi.org/10.1007/978-0-387-32833-1_214

[19] https://arxiv.org/abs/2005.07727

[20] https://arxiv.org/abs/1910.11626

[21] https://arxiv.org/abs/1805.06605

[22] https://arxiv.org/abs/1703.05921

[23] https://openaccess.thecvf.com/content_CVPR_2019/html/Chen_Homomorphic_Latent_Space_Interpolation_for_Unpaired_Image-To-Image_Translation_CVPR_2019_paper.html

[24] https://openaccess.thecvf.com/content_ICCV_2019/html/Abdal_Image2StyleGAN_How_to_Embed_Images_Into_the_StyleGAN_Latent_Space_ICCV_2019_paper.html

https://ieeexplore.ieee.org/document/9008515