Statistical Distance Latent Regulation Loss for Latent Vector Recovery

Jeongik Cho[1]

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

Abstract

Finding a latent vector that can generate specific data by inverting the generative model is called latent vector recovery(or latent vector projection). When performing gradient descent based latent recovery, the latent vector being recovered may escape the train latent distribution. To prevent this, some papers used latent regulation loss or resampling.

In this paper, assuming that the generative model is trained with IID (Independent and Identically Distributed) random variables, I propose statistical distance latent regulation loss, which uses the distance between distribution followed by train latent random variables, and discrete uniform distribution, which assumes that each element of the latent vector has the same probability, as a latent regulation loss. The statistical distance latent regulation loss considers the correlation between each element of the latent vector, so better latent vector recovery is possible.

In this paper, I compared the performances of latent regulation losses and resampling methods of other papers as well as statistical distance latent regulation losses using several statistical distances.

In conclusion, the performances of Wasserstein distance latent regulation loss and Energy distance latent regulation loss were the best.

Also, in this paper, when performing latent vector recovery with a generator trained with an IID random variable, I propose the latent distribution goodness of fit test, an additional test to check whether all elements of all recovered latent vectors follow the distribution of the train latent random variable.

1. Statistical Distance Latent Regulation Loss

The generative model (generator) G is trained to convert the d_z-dimensional multivariate random variable $Z \in R^{d_z}$ following a certain distribution to the d_x-dimensional multivariate random variable $X \in R^{d_x}$. In case of GAN, usually train latent vector $Z \sim U(a,b)^{d_z}$ or $\sim N(\mu,\sigma^2)^{d_z}$, in case of VAE, each element of train latent random variable Z follows a normal distribution with different mean and variance. In this case, finding an ideal latent vector z^* that can generate any data x sampled from a data random variable X using a pre-trained generator G is called latent vector recovery.

There are gradient descent-based and encoder-based methods for latent vector recovery. The encoder-based method requires additional encoder training. In this paper, only the gradient descent-based method is covered.

The gradient descent-based latent vector recovery receives the error between $G(z_p)$, the data generated through latent vector z_p, and the received data x as reconstruction loss, and performs gradient descent repeatedly for the latent vector z_p to reduce reconstruction loss. The following function shows the process of gradient descent-based latent vector recovery.

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

$z_p \leftarrow initialize(\,)$

$repeat\ t\ times$:

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

$L \leftarrow L_{rec}$

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

$return\ z_p$

$initialize$ is a function that initializes the values of z_p. t is the number of times to perform gradient descent. opt is an optimizer. $diff$ is a function that measures the difference between two data. L_{rec} is reconstruction loss. L is the total loss. Through the above function, it can be found the latent vector z_p that minimizes reconstruction loss L_{rec}.

However, when there is a z_p that minimizes reconstruction loss L_{rec}, the obtained z_p is not always an ideal latent vector z^*. The reason is that there is a possibility that z_p is a latent vector sampled from the unexpected latent random variable $K \nsim Z$.

For example, suppose that in the MNIST handwriting data, x is the handwriting data of the number one, $G(z_p)$ currently produces the number zero, and $z_p[1]$, the first element of z_p, represents the width of the letter. If the other elements of the latent vector z_p remain unchanged and $z_p[1]$ becomes extremely low, the width of the character becomes very narrow, so it may look like the number one. The latent vector z_p at this time is a local optima with a sufficiently low reconstruction loss L_{rec}.

However, z_p at this time is a latent vector sampled from the unexpected latent random variable $K \nsim Z$. Also, since the generative model G is not trained to generate out-of-distribution data, there is always a tendency to generate data distribution X. Therefore, the generative model G tends to convert unexpected latent random variable $K \nsim Z$ to data distribution X. Therefore, there may be several global optima latent vector z_p that minimize reconstruction loss L_{rec}. However, among the latent vector z_p, the z_p sampled from the unexpected latent random variable K, not the train latent random variable Z, cannot be the ideal latent vector z^*. This means that an additional term is needed so that $P(Z = z_p)$ can be maximized.

To maximize $P(Z = z_p)$, latent regulation loss was added to loss L in the paper [3, 4], and part of the element of z_p was resampling in the paper [5, 1] after gradient descent. The following function shows latent vector recovery using latent regulation loss.

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

$z_p \leftarrow initialize()$

$repeat\ t\ times:$

$$L \leftarrow diff\left(x, G(z_p)\right) + \lambda_{lr} L_{lr}$$

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

$return\ z_p$

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

In this paper, assuming that the generative model G is trained with IID random variables Z, I propose statistical distance latent regulation loss, which uses the distance between any distribution A followed by train latent random variables $Z \sim A^{d_z}$, and discrete uniform distribution S, which assumes that each element of the latent vector z_p has the same probability (probability mass function $P_S(x) = \begin{cases} \frac{1}{d_z} & if\ x \in z_p \\ 0 & otherwise \end{cases}$), as a latent regulation loss L_{lr}.

Since the statistical distance latent regulation loss can consider the relationship between each element of z_p, a better latent vector z_p can be found. The statistical distance latent regulation loss is as follows.

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

$Dist$ is a function that represents 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 even distribution made from latent vector z_p.

Among the various statistical distances, this paper used four statistical distances: Bhattacharyya distance, Wasserstein distance, Energy distance, and Lukaszyk Karmowski distance. The following table shows the required conditions and features by latent regulation loss or resampling method.

Name	Z~ALL	Z~IID	Z~N	Z~U	Remarks
Bhattacharyya distance			O		
Wasserstein distance		O	O	O	
Energy distance		O	O	O	
Lukaszyk Karmowski distance		O	O	O	
Trick discriminator	O	O	O	O	Hard to find hyperparameter, slow speed
Z score square			O		
Z score absolute			O		
Logistic cutoff			O		Information lost
Truncated normal cutoff			O		Information lost
Boundary resampling				O	Information lost, no hyperparameter

Table 1. Features by method

Z~ALL in the above table means that the train latent vector Z can be used regardless of any distribution, and Z~IID means that Z can be used when following the IID distribution. The yellow items in the table are not suggested in other papers. The "trick discriminator" is the loss proposed in [3]. Z score square is the loss suggested in [4]. The logistic cutoff and truncated normal cutoff are the resampling methods proposed in [5]. Boundary resampling is a resampling method proposed in [1]. Resampling schemes cause information loss when resampling and convergence is slowed down.

2. Latent distribution goodness of fit test

As explained previously, latent vector z_p with low reconstruction loss L_{rec} is not always the ideal latent vector z^*. To check whether the latent vector z_p was sampled from the train latent random variable Z, this paper proposes a latent distribution goodness of fit test.

In [8], the goodness of fit test was used to evaluate the GAN, but in this paper, it is used to verify that the correct latent vector has been recovered.

Assuming that the train latent random variable Z is an IID random variable that follows the random distribution A^{d_z}, the distribution of all elements of the recovered latent vector z_p will follow distribution A. Latent goodness of fit test verifies that the distribution of all elements of all recovered latent vectors follows distribution A. If the latent vectors do not pass the latent goodness of fit test, the latent vectors are not considered to have been properly recovered. However, passing the Latent distribution goodness of fit test does not indicate that latent vectors have been properly recovered. Reconstruction loss L_{rec} is still important. Latent distribution goodness of fit test is an additional test to ensure that latent vector z_p, which minimizes reconstruction loss L_{rec}, is correctly recovered.

3. Experiments

For the experiment, pre-trained GAN using adversarial loss of LSGAN [6] was used. latent vector dimension d_z=256. MNIST handwriting dataset was used. For evaluation, a latent distribution goodness of fit test, L1 loss, L2 loss, and a classifier classification test with an accuracy of 99.3% were used. z_p initialize function $initialize()$ is $sampling(Z)$. The latent vector z_p with the lowest loss L was selected by initializing and optimizing 16 latent vectors per data in parallel. For $diff$, the l_1 $loss$ s with the best result in [2] was used. The gradient descent iteration number $t = 200$ and optimizer $opt = Adam$. For evaluation, only 1000 randomly selected from 10000 test data were used. As the latent distribution goodness of fit test, KS-test (Kolmogorov–Smirnov test) was used. Test is a two-sided test and a $significance\ level = 0.05$. Wasserstein distance, Energy distance, and Lukaszyk Karmowski distance were measured by sampling enough samples (10000) from the train latent random variable Z.

Logistic cutoff and truncated normal cutoff were excluded from the experiment due to too low performance and difficult hyperparameter search. Trick discriminator was also excluded due to its low performance and slow speed. The following tables show the performance according to latent regulation loss when train latent random variable $Z \sim N(0, 1^2)^{d_z}$. GAN's FID is 6.135317.

No regulation		Learning rate		
		0.0001	0.001	0.01	0.1
Goodness of fit test		Pass	Pass	Fail	Fail
Latent mean		0.000777	0.002972	-0.00437	-0.04643
Latent variance		0.989036	1.000564	1.326683	19.93552
L1 loss per pixel		121.0421	39.07714	18.789	20.72877
L2 loss per pixel		12.22913	5.106534	2.63666	2.876963
Classifier accuracy		0.652	0.958	0.987	0.979

Table 2. Without regulation loss

When the $learning\ rate = 0.001\ or\ 0.01$, the latent vector was not significantly different from the initial latent vector due to the learning rate that was too low, so the Goodness of fit test passed, but the L1 loss and L2 loss were high, and classifier accuracy was low. That means L_{rec} is too large.

When $learning\ rate = 0.01\ or\ 0.1$, L_{rec} is considered to be sufficiently low because L1 loss, L2 loss, and classifier accuracy are low, but it is difficult to say that latent vector recovery was properly performed because it failed in the latent distribution goodness of fit test.

Because latent regulation loss lowers latent variance, subsequent experiments experimented with a $learning\ rate = 0.01$ where a latent variance slightly greater than 1 was measured.

Wasserstein distance		Regulation loss weight				
		0.001	0.01	0.1	1	10
Goodness of fit test		Fail	Fail	Pass	Pass	Pass
Latent mean		0.0010	-0.0039	0.0000	0.0000	-0.0003
Latent variance		1.3077	1.1791	0.9996	0.9946	0.9951
L1 loss per pixel		18.8311	18.8883	20.0490	27.5913	71.3795
L2 loss per pixel		2.6464	2.6462	2.7840	3.7509	8.2979
Classifier accuracy		0.9930	0.9920	0.9930	0.9840	0.8500

Table 3. Wasserstein latent regulation loss results

Energy distance		Regulation loss weight			
		0.01	0.1	1	10
Goodness of fit test		Fail	Pass	Pass	Pass
Latent mean		-0.0042	0.0003	-0.0001	0.0002
Latent variance		1.2611	1.0353	0.9971	0.9953
L1 loss per pixel		18.8818	18.9611	26.8357	66.9098
L2 loss per pixel		2.6472	2.6427	3.6495	7.9251
Classifier accuracy		0.9940	0.9890	0.9870	0.8870

Table 4. Energy latent regulation loss results

Wasserstein latent regulation loss and energy latent regulation loss passed the latent distribution goodness of fit test and showed good performance.

Z score square		Regulation loss weight			
		0.001	0.0032	0.0057	0.01
Goodness of fit test		Fail	Fail	Fail	Fail
Latent mean		-0.0035	-0.0022	-0.0033	-0.0039
Latent variance		1.2639	1.1299	0.9932	0.8159
L1 loss per pixel		18.7818	19.0186	18.5449	18.1677
L2 loss per pixel		2.6250	2.6467	2.5950	2.5443
Classifier accuracy		0.9870	0.9880	0.9870	0.9880

Table 5. Z score square latent regulation loss results

Bhattacharyya distance		Regulation loss weight					
		0.01	0.032	0.038	0.043	0.057	0.1
Goodness of fit test		Fail	Fail	Fail	Fail	Fail	Fail
Latent mean		0.0001	-0.0038	-0.0001	0.0014	-0.0025	-0.0028
Latent variance		1.2353	1.054062	1.011107	0.9768	0.893201	0.667697
L1 loss per pixel		18.4579	18.7139	18.6379	18.3467	18.58446	17.82391
L2 loss per pixel		2.5823	2.6216	2.5896	2.5807	2.58473	2.461706
Classifier accuracy		0.9890	0.9880	0.9870	0.9910	0.991	0.983

Table 6. Bhattacharyya latent regulation loss results

Lukaszyk karmowski distance		Regulation loss weight			
		0.01	0.018	0.032	0.1
Goodness of fit test		Fail	Fail	Fail	Fail
Latent mean		-0.0068	-0.0016	-0.0012	0.0012
Latent variance		1.1528	1.033803	0.847241	0.3438
L1 loss per pixel		18.5023	18.2469	18.0396	18.1837
L2 loss per pixel		2.5925	2.5601	2.5294	2.5047
Classifier accuracy		0.9910	0.9920	0.9930	0.9860

Table 7. Lukaszyk karmowski distance latent regulation loss results

Z score absolute		Regulation loss weight				
		0.01	0.014	0.018	0.032	0.1
Goodness of fit test		Fail	Fail	Fail	Fail	Fail
Latent mean		0.0000	0.0011	-0.0015	-0.0003	8.17E-04
Latent variance		1.0946	1.020341	0.936547	0.7215	0.215937
L1 loss per pixel		18.3825	18.6775	18.4642	18.9814	20.91224
L2 loss per pixel		2.5955	2.6094	2.5748	2.6310	2.825397
Classifier accuracy		0.9970	0.9880	0.9900	0.9900	0.993

Table 8. Z score absolute latent regulation loss results

On the other hand, all other latent regulation losses did not pass the latent distribution goodness of fit test, although the latent regulation loss weight was properly adjusted so that the latent mean was 0 and the latent variance was 1. This means that the Wasserstein latent regulation loss or energy latent regulation loss should be used as the latent regulation loss.

The following tables show the performance according to latent regulation loss when train latent random variable $Z \sim U(-1,1)^{d_z}$. GAN's FID is 5.693037.

Wasserstein distance		Regulation loss weight			
		0.01	0.1	1	10
Goodness of fit test		Fail	Fail	Pass	Pass
Latent mean		-0.0062	-0.0002	0.0000	-0.0001
Latent variance		0.4959	0.345214	0.333269	0.3334
L1 loss per pixel		17.5927	18.1273	22.6696	41.2313
L2 loss per pixel		2.4783	2.5142	3.0801	5.3009
Classifier accuracy		0.9830	0.9930	0.9940	0.9660

Table 9. Wasserstein latent regulation loss results

Energy distance		Regulation loss weight		
		0.1	1	10
Goodness of fit test		Fail	Pass	Pass
Latent mean		-0.0006	0.0000	0.0002
Latent variance		0.3766	0.334025	0.333392
L1 loss per pixel		18.0478	23.3761	48.6961
L2 loss per pixel		2.5033	3.1750	6.1208
Classifier accuracy		0.9850	0.9870	0.9260

Table 10. Energy latent regulation loss results

4. References

[1] Zachary C. Lipton, Subarna Tripathi,

"Precise Recovery of Latent Vectors from Generative Adversarial Networks"

https://arxiv.org/abs/1702.04782

[2] Arun Patro ; Vishnu Makkapati ; Jayanta Mukhopadhyay

Evaluation of Loss Functions for Estimation of Latent Vectors from GAN

https://ieeexplore.ieee.org/document/8517097/authors#authors

[3] Raymond A. Yeh, Chen Chen, Teck Yian Lim, Alexander G. Schwing, Mark Hasegawa-Johnson, Minh N. Do

Semantic Image Inpainting with Deep Generative Models

https://arxiv.org/abs/1607.07539

[4] Antonia Creswell, Anil A Bharath

Inverting The Generator Of A Generative

Adversarial Network (II)

https://arxiv.org/abs/1802.05701

[5] Nicholas Egan, Jeffrey Zhang, Kevin Shen

Generalized Latent Variable Recovery for Generative Adversarial Networks

https://arxiv.org/abs/1810.03764

[6] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, Stephen Paul Smolley

Least Squares Generative Adversarial Networks

https://arxiv.org/abs/1611.04076

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

THE MNIST DATABASE of handwritten digits

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

[8] Ryan Webster, Julien Rabin, Loic Simon, Frederic Jurie

Detecting Overfitting of Deep Generative Networks via Latent Recovery

https://arxiv.org/abs/1901.03396