Wasserstein 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 using a generative model is called latent vector recovery. When performing gradient descent based latent recovery, the latent vector being recovered may escape the train latent vector distribution. To prevent this, latent regulation loss has been used in many papers. In this paper, I propose a Wasserstein latent regulation loss to improve the performance of latent recovery, assuming that the generative model is trained with IID (Independent and identically distributed) random variables. The proposed Wasserstein latent regulation loss is the Wasserstein distance between the sample distribution of the train probability distribution and the latent vector being recovered. This paper compares the latent regulation loss of several papers, including the proposed Wasserstein latent regulation loss. In conclusion, the Wasserstein regulation loss and the log normal density function proposed in [1] showed the best performance.

1. Latent regulation loss

The generative model (generator) G is trained to convert the d_z dimension multivariate random variable $Z \in R^{d_z}$ that follows a specific distribution to the d_x dimension data multivariable random variable $X \in R^{d_x}$. Finding a latent vector z_p that can generate specific data x_p sampled from X through gradient descent using a pre-trained generator G is called gradient descent based latent vector recovery. In this paper, only latent vector recovery using gradient descent is covered.

The following function is a function that performs gradient descent based latent vector recovery.

```
$function\ latent\_recovery(x_p, G, n, opt)$:
    $z_p \leftarrow sampling(Z)$
    $repeat\ n\ times$:
        $L \leftarrow diff\left(x_p, G(z_p)\right)$
        $z_p \leftarrow z_p - opt\left(\frac{\Delta L}{\Delta z_p}\right)$
    $return\ z_p$
```

$sampling$ is sampling one value from a random variable. n is the number of times to perform gradient descent. opt is an optimizer. $diff$ is a function that measures the difference between two data.

In this paper, I propose a method of adding Wasserstein latent restriction loss to loss L to improve the performance of latent recovery when Z is an IID random variables that follows an specific probability distribution A^{d_z}.

When $Z \sim A^{d_z}$, the set of each element of $sampling(Z)$ can be called the one-dimensional sample distribution $\bar{A}$ sampled d_z times from probability distribution A. Similarly, each set of elements of z_p is treated as a sample distribution, and the Wasserstein distance between the two sample distributions becomes a Wasserstein latent restriction loss.

```
$function\ latent\_recovery(x_p, G, n, opt)$:
    $z_p \leftarrow sampling(Z)$
    $repeat\ n\ times$:
        $L_{lr} \leftarrow W_{dist}\left(z_p, sampling(Z)\right)$
        $L \leftarrow diff\left(x_p, G(z_p)\right) + \lambda_{lr} L_{lr}$
        $z_p \leftarrow z_p - opt\left(\frac{\Delta L}{\Delta z_p}\right)$
    $return\ z$
```

W_{dist} is the Wasserstein distance between the two sample distributions. Reducing the latent restriction loss is to increase $P(Z = z_p)$. z_p with high $P(Z = z_p)$ is also high in $P\left(X = G(z_p)\right)$, which means that if generator G is ideal, z_p not only expresses x_p well, but also has a low probability of falling into local optima. For example, suppose that in MNIST handwriting data, x_p is the handwriting of the number "1" picture, $G(z_p)$ produces the current number "0" picture, and $z_p[1]$ represents the width of the letter. If the other elements of z_p remain unchanged and $z_p[1]$ becomes extremely low, the width of the letter becomes very narrow, which can look like the number "1" picture. However, for z_p where $z_p[1]$ is extremely low, $P(Z = z_p)$ will be very low or zero. In this case, z_p cannot be regarded as a good latent vector that can generate x_p, and since it has already converged to local optima, it is highly likely that even if additional gradient descent is performed, global optima number "1" picture cannot be generated. That is, good z_p can be regarded as a value that minimizes $diff\left(x_p, G(z_p)\right)$ while maximizing $P(Z = z_p)$.

In general, since GAN is trained with IID random variables such as $Z \sim U(a, b)^{d_z}$ or $Z \sim N(\mu, \sigma^2)^{d_z}$, proposed Wasserstein latent restriction loss can be used for latent vector recovery. In addition, in the case of VAE, since all elements of the latent vector follow a normal distribution, it can be transformed into an IID random variable through scale and shift.

2. Experiments

For the experiment, pre-trained GAN using LSGAN [6] adversarial loss was used. MNIST handwriting dataset [7] was used for the data. For the performance evaluation, I measured how well the generated data were classified into a classifier with an accuracy of 99.4%. For $diff$, the l_1 loss with the best result in [2] was used. The number of gradient descent iterations is $n = 200$ and optimizer $opt = Adam$. For evaluation, only 1000 of 10000 test data were used.

The latent regulation loss compared is seven, including the two distances added in this paper.

"Wasserstein distance" and "Energy distance" are added latent regulation loss in this paper. Energy latent regulation loss uses the energy distance between two sample distributions. The "trick discriminator" is the loss proposed in [3].

The log normal density is the loss suggested in [4]. The logistic cutoff and truncated normal cutoff are the clipping methods proposed in [5]. Logistic cutoff has two hyperparameters, of which b is fixed to 2, which showed the best performance in the paper.

The following table shows the accuracy of the classifier according to the learning rate and hyperparameter.

Wasserstein distance		Regulation loss weight				
		0	0.01	0.1	1	10
Learning rate	0.001	46	47.1	45.1	37	19.7
	0.01	75.8	79.5	80.5	70.2	46.8
	0.1	73.5	83.9	94.3	**94.7**	86.6
	1	49.4	74.2	82.2	52.5	41.4

Table 1. Wasserstein distance accuracy

Energy distance		Regulation loss weight				
		0	0.01	0.1	1	10
Learning rate	0.001	46	49.4	46.4	41	23.5
	0.01	75.8	77.9	79	72.1	47.3
	0.1	73.5	74.5	92.6	**93.2**	85.1
	1	49.4	55.4	66.8	81.3	69.2

Table 2. Energy distance accuracy

Trick discriminator		Regulation loss weight					
		0	0.000001	0.00001	0.0001	0.001	0.01
Learning rate	0.001	46	47.9	48.2	45.3	45.1	35
	0.01	75.8	77.6	**77.8**	77.5	77	66.7
	0.1	73.5	73	69.1	71.3	69.3	54.8
	1	49.4	46.9	45.9	49	51.2	40.2

Table 3. Trick discriminator accuracy

Log normal density		Regulation loss weight		

		0	0.01	0.1	1	10
Learning rate	0.001	46	46.2	46.3	43.6	27.4
	0.01	75.8	78.9	87.2	**95.5**	91.9
	0.1	73.5	93	95.1	86.9	69.9
	1	49.4	15.4	58.3	33.7	20.9

Table 4. Log normal density accuracy

Logistic cutoff		hyperparameter			
		-1	0	1	2
Learning rate	0.01	49.32	10.3	13.5	49.28
	0.1	73.3	25	52.1	79.1
	1	47.2	**77.9**	71.3	70.9
	10	31.6	76.6	23.6	16.1
	100	22.5	73.8	8.5	9.9

Table 5. Logistic cutoff accuracy

Truncated normal cutoff		Hyperparameter			
		0	1	2	3
Learning rate	0.01	10.9	15.2	51	83.9
	0.1	22	53.1	80.8	**87.5**
	1	75.1	69.9	76.5	77
	10	76.5	24.5	15.5	16.2
	100	74.6	9.8	10.5	11.5

Table 6. Truncated normal cutoff accuracy

3. Conclusion

Overall, the performances of Wasserstein latent regulation loss and Log normal density latent regulation loss were the best.

4. References

[1] Antonia Creswell, Anil A Bharath,

Inverting The Generator Of A Generative Adversarial Network (II),

https://arxiv.org/abs/1802.05701

[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/