GAN Using Capsule Network for Discriminator and Generator

Kanako Marusaki
NEC Corporation
Tokyo, Japan
k-marusaki@suou.waseda.jp

Hiroshi Watanabe
Graduate School of Fundamental Science and Engineering
Waseda University
Tokyo, Japan
hiroshi.watanabe@waseda.jp

***Abstract*— In this paper, we propose Capsule GAN, which incorporates the capsule network into the structure of both discriminator and generator of Generative Adversarial Networks (GAN). Many CNN-based GANs have been studied. Among them, Deep Convolutional GAN (DCGAN) has been attracting particular attention. Other examples include convolutional GAN, auxiliary classifier GAN, Wasserstein GAN (WGAN) which uses Wasserstein distance to prevent mode collapse during the learning process, and Wasserstein GAN-gp (WGAN-gp). However, image generation by GAN is not stable and prone to mode collapse. As a result, the quality of the generated images is not satisfactory. It is expected to generate better quality images by incorporating a capsule network, which compensates for the shortcomings of CNN, into the structure of GAN. Therefore, in this paper, we propose two approaches to generate images with better quality by incorporating the capsule network into GAN. The experimental results show that the proposed method is superior to the conventional method.**

***Keywords*— capsule network, gan, deep learning, cnn**

I. INTRODUCTION

In recent years, many image processing methods using Convolutional Neural Networks (CNNs) [1] have been proposed. Generative Adversarial Network (GAN) [2], which is an adversarial generative network that generates new data, is one of them, and CNNs are often used. However, image generation by Generative Adversarial Network (GAN) is difficult and the quality of the generated images is not stable.

Hinton's group proposed the capsule network in 2017 [3]. This approach compensates for the shortcomings of CNNs, and is expected to generate better quality data by incorporating the capsule network into GAN.

In this paper, we propose "Capsule GAN", which incorporates the capsule network into the structure of both discriminator and generator of GAN. Our approach has two versions: a simple version that transforms one layer of the discriminator into the generator, and a version that fully trains the capsule network in both the discriminator and the generator.

II. RELATED WORKS

GAN is a model that uses two networks called discriminator and generator to generate data similar to the input data. Fig.1 shows a schematic diagram of a basic GAN, where the generator takes a random number as input and generates data similar to the dataset and outputs it. The discriminator takes the data generated by the generator and the training dataset as input, and discriminates whether the input data is the training dataset (real) or the generated data (fake). The discriminator learns to identify the input data in such a way that it is not fooled by the generator. In other words, the discriminator and the generator learn by competing with each other.

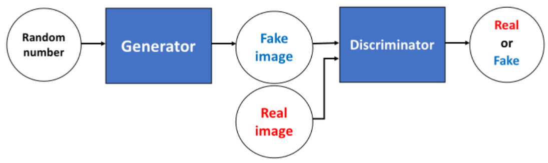


Fig. 1. Schmatic diagram of a basic GAN.

The work of Gadirov [4] and Jaiswall [5] are two examples of GANs that use capsule network. These two GANs use capsule network only for the discriminator and CNN for the generator. In this paper, we define the GAN proposed by Gadirov as Capsule GAN1 since the author provides codes for the algorithm. In their papers, the reason not to use capsule network for generator is not clear. The structure of capsule network is shown in Fig. 2.

Capsule GAN1 successfully generates images with higher quality than Deep Convolutional GAN (DCGAN) [8], which is a typical CNN-based GAN in MNIST [6] and CIFAR-10 [7]. However, the target images are limited to simple ones, and the quality for more complex images is not clear. The inherent advantage of the capsule network is that it can preserve the mutual positioning of parts in recognizing objects with complex shapes.

III. PROPOSED METHOD

A. Structure using one discriminator layer in a Generator (Capsule GAN2)

We attempt to apply the capsule network to the generator as well. In this structure, we use the capsule network for both the discriminator and the generator. In the generator, we use a DigitCaps layer that holds the image features extracted from the discriminator. The DigitCaps layer is generated from the capsule network, but it is embedded in the generator just like a regular CNN. The input to the generator is the extracted DigitCaps layer multiplied by a random number. In this paper, we refer to this capsule GAN as "Capsule GAN2." The detailed structure this method is shown in Fig. 3.

The DigitCaps layer is used only when the image features of the training dataset are input. The flow of using the DigitCaps layer is shown in Fig. 4. It shows an example of the process when x images are input.

B. Structure using capsule network as a generator (Capsule GAN3)

In this structure, capsule network fully is used for both the discriminator and the generator. Similar to Capsule GAN2, the generator incorporates the structure that reverses the flow of capsule network. In this paper, we refer this Capsule GAN as

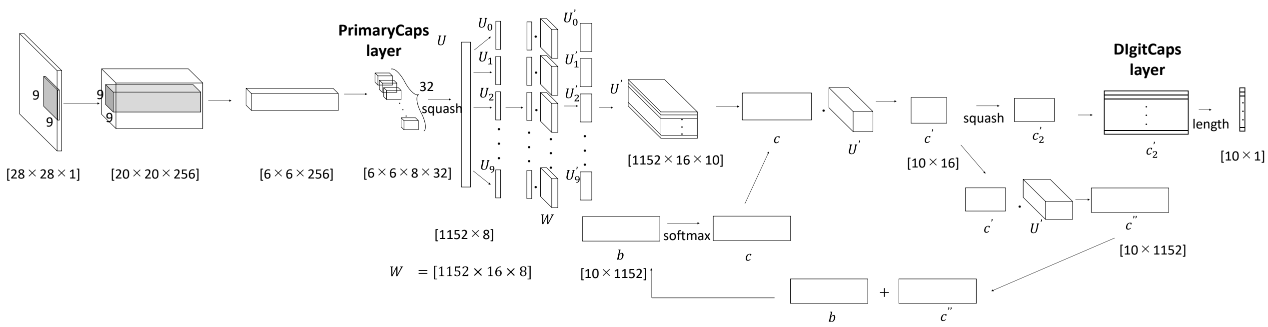


Fig. 2. Structure of the capsule network.

"Capsule GAN3," which also proposes a structure to generate images for each generator class using the labels of the generator classes. Fig. 5 shows the generator structure of Capsule GAN3 when one weight matrix is shared. The structure shown in Fig. 5 is for generating a 28x28 pixels image. The structure of generator is the same as Capsule GAN2. It increases the values that are important for generating images by dynamic routing before inputting them to the deconvolution layer. As a result, the quality of the generated image can be stabilized.

Fig. 6 shows the structure of the generator of Capsule GAN3 when the weight matrix is used for each generation class. The structure shown in Fig. 6 is the same as Fig. 5, but for generating 28x28 pixels images. The generated classes are 10 classes. The flow after the matrix U' is the same as in Fig. 5. A different weight matrix is used for each generation class. Therefore, a label indicating the class of the dataset is used during training. As a result, Capsule GAN3 has a structure that incorporates Capsule Network in both discriminator and generator like a CNN.

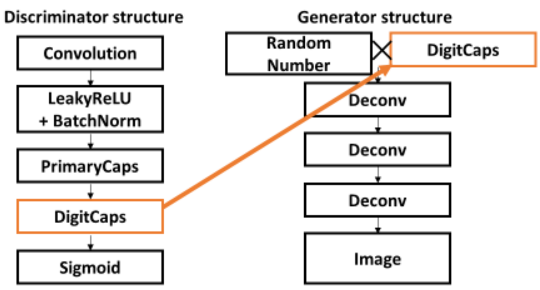


Fig. 3. Detailed structure of Capsule GAN2.

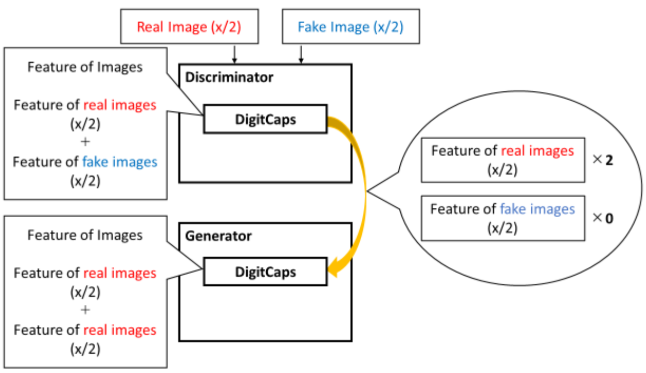


Fig. 4. DigiCaps layer usage in Capsule GAN2.

IV. EXPERIMENT

A. *Performance comparison*

In the experiment, we compared the four types of GANs: CNN-based GAN, Capsule GAN1, Capsule GAN2, and Capsule GAN3. The schematic diagrams of the GANs used in this experiment are shown in Fig. 7.

The datasets are MNIST and FashionMNIST [9] for black and white images and cat images for color images. For the cat images, images are generated using the Wasserstein GAN-gp (WGAN-gp) [10] method, which is known to be an effective stabilization method for GANs. Therefore, the GAN using the compared CNNs is WGAN-gp. The results of the generated cat images are shown in Fig. 8.

The Inception Score (IS) [11] and Geometry Score (GS) [12] are used to evaluate the quality of generated images in the experiment. The obtained values are shown in Table I. From Fig. 8, we can see that Capsule GAN2 and Capsule GAN3 are able to capture the cat's shape better than the other two results in the visual comparison. Table I also shows the Capsule GAN3 has the best evaluation score in all datasets. Therefore, it is regarded that Capsule GAN3 is able to generate the best quality images among the four types of GANs that we compared.

B. *Verification experiment using labels*

In the verification experiment, we verified whether images can be generated for each class in the structure of Capsule GAN3. Fig. 9 shows the results of MNIST generation, where each row has a different label and three examples of generated images. From Fig. 9, we can see that we are able to generate images for each class indicated by the generated class label. Three examples have different appearances but are recognized as the same label.

TABLE I. EVALUATED RESULTS OF GENERATED IMAGES BY INSEPTION SCORE (IS) AND GEOMETRY SCORE (GS)

Data Set	Method	GAN (Using CNN)	Caps. GAN1	Caps. GAN2	Caps. GAN3
MNIST	IS	2.32	2.35	2.37	**2.57**
	GS (*100)	3.89	5.01	3.48	**1.46**
Fashion MNIST	IS	4.39	4.34	4.48	**4.54**
	GS (*100)	0.12	0.13	0.21	**0.10**
Cat	IS	4.17	4.53	4.68	**4.72**
	GS (*100)	0.19	0.22	3.84	**0.11**

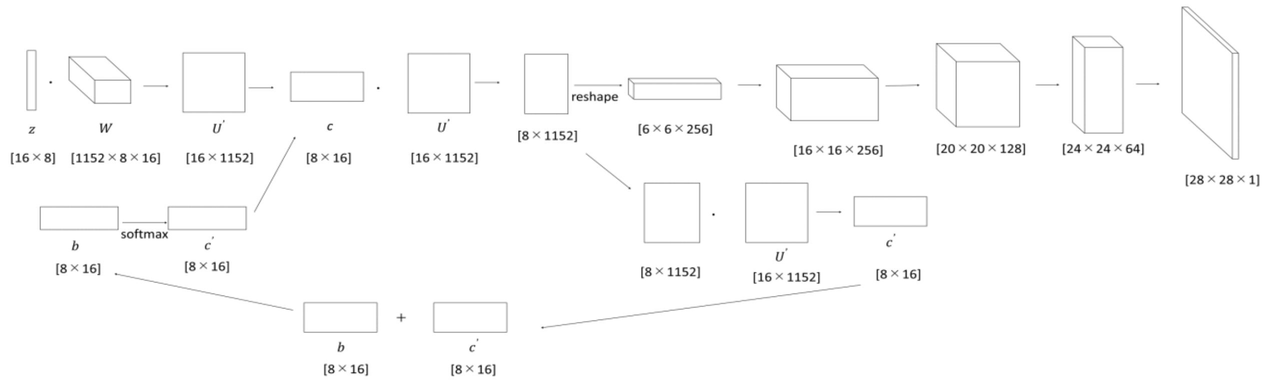


Fig. 5. Generator structure of Capsule GAN3. (when sharing one weight matrix).

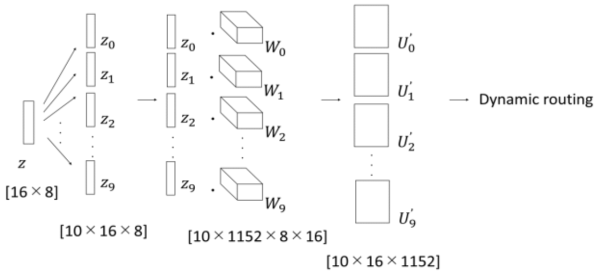


Fig. 6. Generator structure of Capsule GAN3. (when sharing multiple weight matrices).

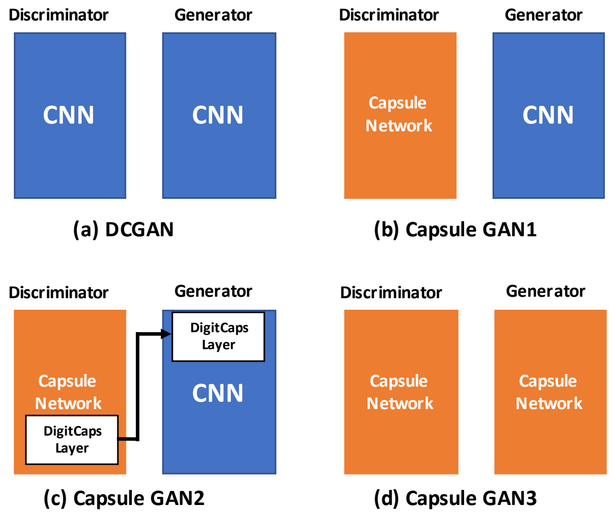


Fig. 7. Schematic diagram of GANs, (a) Conventional DCGAN, (b) Capsule GAN1[4], (c) Capsule GAN2, and (d) Capsule GAN3.

V. CONCLUSION

In this paper, we proposed Capsule GAN2 and Capsule GAN3, which use capsule network as discriminator and generator. Capsule GAN2 use only DigitCaps layer in the generator, so that the quality improvement of generate images is somewhat limited. Capsule GAN3, which employs full operation of capsule network in the generator, produces the best quality images in all datasets used in the experiments. In addition, we confirmed that Capsule GAN3 can generate images for each generation class by using labels.

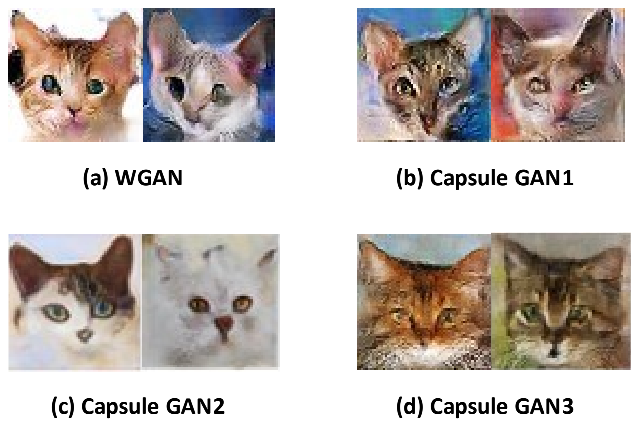


Fig. 8. Examples of generated cat images for (a) WGAN, (b) Capsule GAN1, (c) Capsule GAN2, and (d) Capsule GAN3.

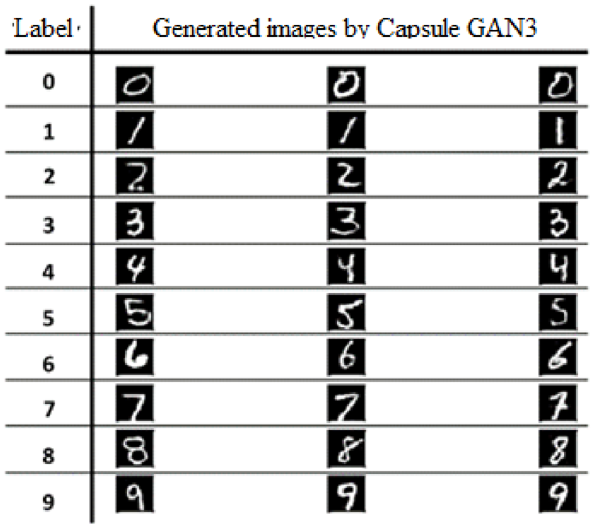


Fig. 9. Verification results of the perofrmance of Capsule GAN3. (Three generated images are shown as examples.).

VI. APPENDIX A

Generated MNIST images are shown in Fig. 10 (a) Conventional DCGAN, (b) Capsule GAN1, (c) Capsule GAN2, and (d) Capusule GAN3. Although the qualitative quality evaluation depends on personal views, we can at least say that the proposed method, especially Capsule GAN3, does not cause shape collapse as seen in DCGAN.

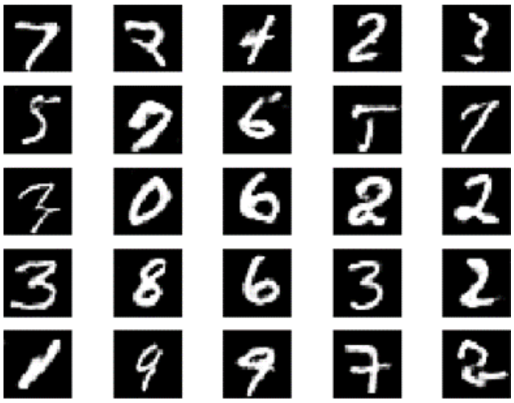
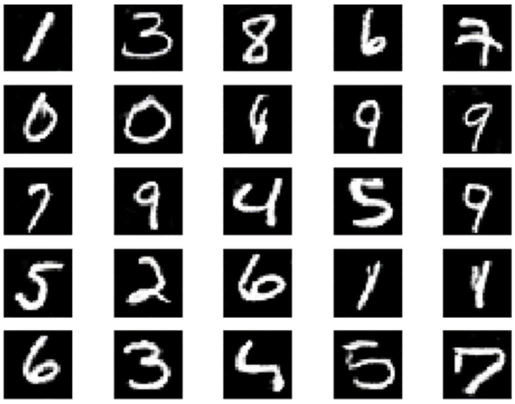


Fig.10 (a) Generated MNIST images by the conventional DCGAN.



.10 (b) Generated MNIST images by Capsule GAN1.

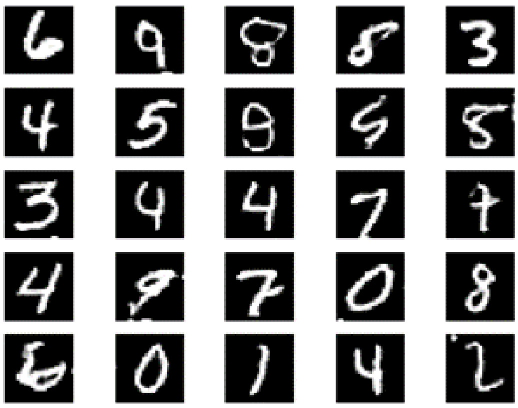


Fig.10 (c) Generated MNIST images by Capsule GAN2. (our proposal).

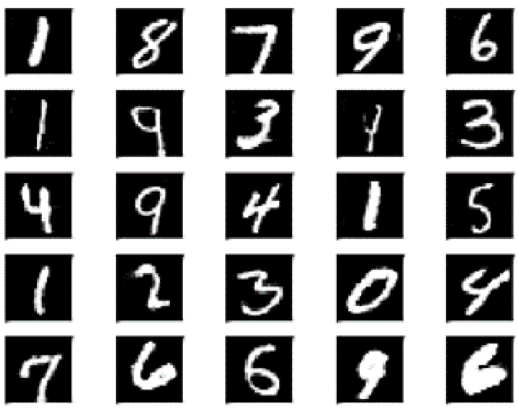


Fig.10 (d) Generated MNIST images by Capsule GAN3. (our proposal).

REFERENCES

[1] A. Krizhevsky, I. Sutskever and G. Hinton, “ImageNet classification with deep convolutional neural networks,” Neural Information Processing Systems (NIPS), pp. 1106-1114, Dec. 2012.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, “Generative adversarial networks,” Neural Information Processing Systems (NIPS), pp. 2672-2680, Dec. 2014.

[3] S. Sabour, N. Frosst and G. E. Hinton, “Dynamic routing between capsules,” Neural Information Processing Systems (NIPS), pp. 3859-3869, Dec. 2017.

[4] H. Gadirov, “Capsule architecture as a aiscriminator in generative adversarial networks,” Master Tthesis, Vytautas Magnus University, Feb. 2018.

[5] A. Jaiswal, W. AbdAlmageed, Y. Wu and P. Natarajan, “CapsuleGAN: Generative adversatial capsule network,” European Conference on Computer Vision (ECCV), pp-526-535, Sep. 2018.

[6] Y. LeCun, C. Cortes and C. J. C. Burges, “The MNIST database of handwritten digits,” http://yann.lecun.com/exdb/mnist/, 1998.

[7] Alex Krizhevsky, “Convolutional deep belief networks on CIFAR-10,” Aug. 2010.

[8] A. Radford, L. Metz and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” International Conference on Learning Representations (ICLR), Jan. 2016.

[9] H. Xiao, K. Rasul and R. Vollgraf, “Fashion-MNIST: A Novel Image Dataset,” arXiv preprint arXiv: 1708.07747, 2017.

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, “Improved training of Wasserstein GANs,” Neural Information Processing System (NIPS), pp. 5769-5779, Dec. 2017.

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, “Improved techniques for training GANs,” Neural Information Processing Systems (NIPS), pp.2234-2242, Dec. 2016.

[12] V. Khrulkov and I. Oseledets, “Geometry Score: A method for comparing generative adversarial networks,” International Conference on Machine Learning (ICML), pp. 2621-2629, Jul. 2018.