Could Spike Neural Networks Beat ANNs on Image Synthesis?

Liyuan Mao*, Yilin Sun*, Haoyu Zhen*
Shanghai Jiao Tong University

Abstract

Spiking Neural Networks (SNNs) have demonstrated their superiority in many applications, such as image classification and semantics segmentation. However, the performance of SNNs on image synthesis is still far behind that of ANNs. In this project, our goal is to explore the possibility of using SNNs to generate images. First, we supplant the generator and the discriminator in GANs with SNNs. Then, we propose an SNN-based Module to extract and embed the features of images. Abundant and concrete experiments are conducted on MNIST and CIFAR-10 datasets.

1. Introduction

Imagination and creativity are two important aspects of human intelligence. Image synthesis is a typical application of these two aspects. Recently, Generative Adversarial Networks (GANs) [1, 4, 7] and Denoising Diffusion Probabilistic Models (DDPMs) [2, 5] have achieved great success in image synthesis. They could generate images that are indistinguishable from real images. Meanwhile, the performance of Spike Neural Networks (SNNs) [3, 6] is still far behind that of Artificial Neural Networks (ANNs). With belief that Spike could reveal the essence of human intelligence, we explore the possibility of using SNNs to generate images. Firstly, we replace the discriminator with an SNN classifier. But when we use an SNN-based generator, the performance becomes worse. To ameliorate this problem, we supplant different parts of the generator with some spike neurons. Finally, we propose a Spike Neural Module to extract and embed the features of images. Furthermore, we conduct abundant and concrete experiments on MNIST and CIFAR-10 datasets.

In summary, our contributions are as follows:

- We embed SNNs Module in to traditional GANs.
- We conduct abundant and concrete experiments on MNIST and CIFAR-10 datasets to illustrate the superiority of SNNs.
- We give a detailed analysis of the results of experiments, aiming to answer the question: Could Spike Neural Networks Beat ANNs on Image Synthesis?

*These authors contributed equally to this work. The list order is random. Liyuan Mao built the whole codebase and implement the baseline model. Yilin Sun implemented all the SNN-based models. Haoyu Zhen lead the project, designed the experiments, and analyzed the results. We concieved the idea of this project together.

2. Methodology

In this section, we will elaborate in detail how we supplant the traditional ANN-based generators and discriminators with the SNN-based corresponding ones. Also, a description of the performance of our models will be given. More image generation results can be found in the experiments section.

2.1. SNN-based Discriminator

Based on our knowledge that SNN models have the capability of image classification, our first idea is to use SNN-based model as discriminator to guide the ANN-based generators. As for image generation tasks, the discriminator is only carrying out a binary classification task, so the spike encodings should be sufficient to provide information for classification.

As it turns out, SNN discriminator does not affect the quality of images generated. Although the images contain more noises, they are still clearly identifiable. Since we have done this experiment, for the upcoming sections, we will remain the usage of SNN discriminator unless otherwise stated.

2.2. Supplanting Generator with SNNs

Having achieved success with the SNN discriminators, we now apply a progressive method to introduce spiking modules gradually to the ANN generators to observe how the performance will be affected.

Our intuition is that the position of spiking modules will largely influence the performance of generators. Since our original ANN generator is a deep convolutional neural network, we would add spiking modules at the front, middle and back of original ANN model.

The structure of our supplanted models is shown in figure 1. The left large blue block indicates the generator as a whole, and the small green blocks indicates the position of where the spiking modules are added to the generator and what those modules are composed of.

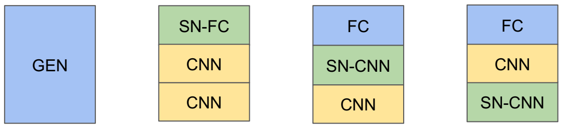


Figure 1. Structures of SNN supplanted Generators

We implemented three parallel generators 1 which have same volume(with respect to parameters and layers) and conducted experiments on these generators to observe their performance. As it turns out, either adding spiking modules at the front or at the back will not generate semantically meaningful pictures. Only by adding spiking modules in the middle of the generator can we get images comparable to the fully ANN-based generators. A detailed illustration and comparison of images will be shown in the experiment section.

2.3. SNN Extractor

The success of supplanting generators with spiking modules inspire us to treat the SNN module as a whole instead of embedding them into traditional ANN generators. We now introduce a new structure of generator 2. In this generator, we fully exploit the potential of SNN spiking outputs by implementing SNN modules as sequence extractor. For any input we want to sample from, we first encode them into the latent space by traditional fully connected layers, then we use a SNN extractor to extract features from the vectors from the latent space. After that, we use convolutional layers to integrate the features and generate actual image.

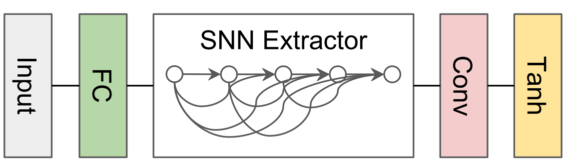


Figure 2. Structure of Generator using SNN Extractor

Our results show that SNN does have the ability of extracting features from latent space. The quality of images are comparable to those generated by embedding spiking modules in the middle of generators.

3. Experiments

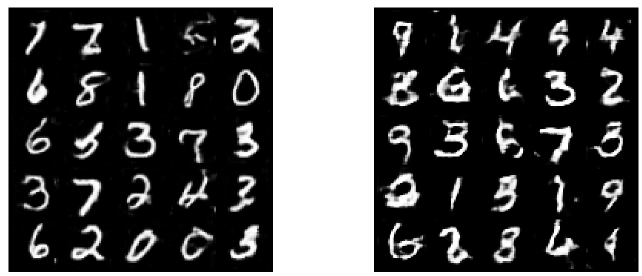
In this section, we will illustrate in detail the results of our experiments, namely, the images generated by various types of generators and discriminators. We will carefully mark the generators and discriminators in use and show some of the selected images they generated.

Note that for any generator we use, there will be two kinds of discriminators, i.e. the ANN and SNN ones. The generators are under our highlight, yet we will always backup with two discriminators for rigorousness.

3.1. MNIST dataset

3.1.1 ANN Generator

We implemented the traditional ANN generator as our baseline for further comparison. This ANN generator is a deep convolutional generator, with three convolutional layers connected by batchnorm and upsampling layers. The results 3 are shown here for the sake of comparison for SNN based generators to come.



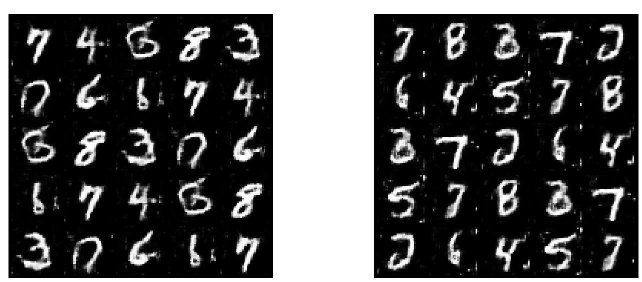
(a) ANN discriminator (b) SNN discriminator

Figure 3. ANN generator results

3.1.2 SNN-supplanted Generator

From our experiments, the middle-part SNN-supplanted generators work well. The images 4 generated are meaningful and clearly recognizable. The drawbacks remain that the images are sort of affected by noises, yet that do not harm the whole quality of images.

Though the middle-part embedding method works, either front or back embedding does not work. The images from these two methods 6 will be provided and analyzed in upcoming sections.

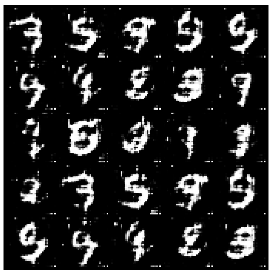


(a) ANN discriminator (b) SNN discriminator

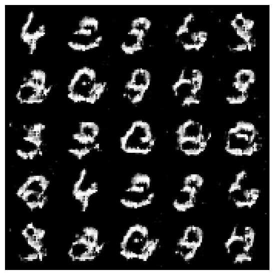
Figure 4. Middle-supplanted generator results

3.1.3 SNN Extractor embedded Generator

The images 5 generated by our generator using SNN extractor are slightly distorted compared to those generated by our previous methods. Yet they do hold semantically meaningful results. The reason why generator with individual SNN extractor module does nor work as well should be that SNN modules are unstable, hence lacking some information during the extracting process.



(a) ANN discriminator

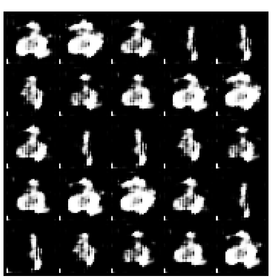


(b) SNN discriminator

Figure 5. SNN extracting generator results

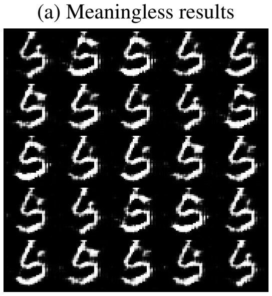
3.1.4 Classic failures with SNN Generators

We also collected some of the classic unsuccessful images 6 from our experiments, especially from those generated with spiking modules added to the front or back of the generators. Moreover, SNN-based generators need proper parameter initialization to function well. Our observation is that SNN modules are highly unstable and the gradients are prone to explode or vanish, hence lossing information during the forward or backward process.

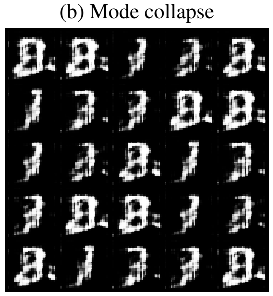


(a) Meaningless results

(b) Mode collapse



(c) Mode stagnation



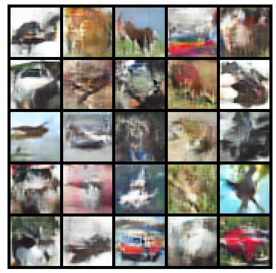
(d) Locally optimal

Figure 6. Classic unsuccessful results

3.2. CIFAR10 dataset

3.2.1 ANN Generator

Same as the MNIST dataset, we still first implemented the ANN generators as our baseline. The structures of networks are identical to those used in previous section, we only modified the output channels.



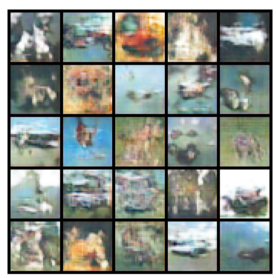
(a) ANN discriminator

(b) SNN discriminator

Figure 7. ANN generator results

3.2.2 SNN-supplanted Generator

As for the SNN-supplanted generators, only those middle-supplanted ones provide meaningful results. Here are some of the selected results of our generators.



(a) ANN discriminator

(b) SNN discriminator

Figure 8. Middle-supplanted generator results

4. Conclusion

We have implemented two paradigms of introducing spiking modules and SNNs into the world of traditional ANNs. Our method of integrating spiking modules with convolutional layers and fully connected layers works well on the image generation task, and our method of assembling SNNs with other modules reveal great potential of SNNs to encoding or extracting features, especially when cooperating with traditional ANNs.

Our insight from abundant and concrete experiments is listed as follows:

- The gradients of SNNs are unstable, hence the performance is prone to be affected by initialization.

- On account of the characteristics that SNNs are less sensitive to noises, the SNNs need to be embedded with fully connected layers before actually extracting features from the inputs.
- Since SNNs generate outputs represented by zero-one sequence, they are not ideal to serve as the last layer of an integrated network. Outputs from SNNs need to be decoded or mapped by convolutional(or other non-spiking) layers to work well.

Could Spike Neural Networks beat ANNs on Image Synthesis? The answer is that SNNs can at least perform equally well as ANNs for now, even though not outperforming ANNs. What is of significance is that the potential of SNNs cannot be neglected. With more research done and more efficient implementation accomplished on SNNs, we believe that SNNs will finally find their place to shine.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. 1

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. 1

[3] Jason Kamran Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu. Training spiking neural networks using lessons from deep learning. *CoRR*, abs/2109.12894, 2021. 1

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. 1

[6] Vineet Kotariya and Udayan Ganguly. Spiking-gan: A spiking generative adversarial network using time-to-first-spike coding. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2022. 1

[7] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. 1