

Импульсные генеративно-состязательные нейронные сети с частотным кодированием

И.Н.Аглюков*, Д.И.Антонов**,***, С.В.Сухов***

*Высшая школа экономики, Москва, Россия

**Ульяновский государственный технический университет, Ульяновск, Россия

***УФИРЭ им. В.А. Котельникова РАН, Ульяновск, Россия

Поступила в редколлегию 29.05.2023

Аннотация—Генеративные методы в искусственных нейронных сетях в настоящее время испытывают взрывообразное развитие. По сравнению с традиционными искусственными нейронными сетями, импульсные нейронные сети (ИмНС) обладают большей энергоэффективностью при решении задач машинного обучения. В то же время методы обучения ИмНС разработаны недостаточно полно, а генеративные методы для ИмНС практически отсутствуют. Данная работа предлагает генеративно-состязательную сеть, кодирующую информацию импульсами (спайками). В сети используются нейроны, обладающие свойством интеграции и возбуждения (integrate-and-fire). Кодирование информации осуществляется в частотном режиме. ИмНС была обучена методом, аппроксимирующим обратное распространение ошибки. Генеративная сеть реализована в пакете SNNToch. Эксперименты, проведенные на общедоступном наборе данных MNIST, показали способность генеративной ИмНС создавать изображения высокого качества.

КЛЮЧЕВЫЕ СЛОВА: импульсная нейронная сеть, генеративная нейронная сеть, генеративно-состязательная сеть

DOI: 10.53921/18195822_2023_23_3_354

ВВЕДЕНИЕ

Два основных подхода в машинном обучении можно определить как дискриминационный и генеративный. В генеративном подходе статистическая модель стремится воспроизвести полную вероятность распределения исходных данных. Генеративные методы в традиционных искусственных нейронных сетях (сетях второго поколения, иногда называемых "аналоговыми" [1], [2]) в настоящее время испытывают взрывообразное развитие. Это связано с разработкой многочисленных приложений, использующих генеративные модели [3]: генерация фотореалистичных изображений, включая фотографии человеческих лиц, преобразование текста в изображение и обратно, получение изображений со сверхразрешением, постановка медицинских диагнозов, разработка новых формул лекарственных средств и т.д. Импульсные (спайковые) нейронные сети (ИмНС) являются новым поколением искусственных нейронных сетей [4], [5], в которых нейроны взаимодействуют с помощью последовательностей коротких импульсов. Эти сети обладают многообещающей перспективой использования из-за их энергоэффективности. ИмНС обладают биологической правдоподобностью и в какой-то мере пытаются воспроизводить нейрональную динамику, присутствующую в мозге [6], [7]. Таким образом, интерес к этим сетям обусловлен, с одной стороны, возможностью моделирования с их помощью динамики нейронных сетей мозга [8], [9], [10]. С другой стороны, ИмНС могут

служить основой для организации вычислительного процесса в перспективном нейроморфном оборудовании [11], [12]. Информация в ИмНС кодируется частотой и временем импульсов (спайков). В связи с этим, обычные методы обучения нейронных сетей (метод обратного распространения ошибки) напрямую к ИмНС неприменимы [1], [13]. Методы обучения ИмНС разработаны недостаточно полно, а генеративные методы для ИмНС практически отсутствуют. В настоящее время известна всего пара работ, посвященных импульсным генеративным сетям. В работе [14] авторы предложили состязательное правило обучения в импульсной генеративной сети, основанное на временном обратном распространении ошибки с суррогатным градиентом. Работа [15] использует локальные правила обучения и использует исходные данные с временным распределением. В данной работе мы предлагаем генеративно-состязательную сеть, составленную из импульсного генератора и аналогового дискриминатора. Обученная на модифицированном алгоритме обратного распространения ошибки, данная сеть способна воспроизводить синтетические изображения хорошего качества.

1. АНАЛОГОВЫЕ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫЕ СЕТИ

Генеративно-состязательные сети (ГСС, в англоязычной литературе Generative Adversarial Networks, GANs) [16], [17], [18], наряду с автоэнкодерами [19], в настоящее время являются одними из наиболее многообещающих и наиболее исследуемых архитектур генеративных сетей. Архитектура ГСС построена на основе двух нейронных сетей-антагонистов – генератора (G , англ. Generator) и дискриминатора (D , англ. Discriminator). Принцип действия ГСС состоит в том, что генератор создаёт образцы на основе исходного набора данных, а дискриминатор пытается отличить сгенерированные образцы от подлинных. В своей работе генератор преобразует сигналы известного распределения (например, нормального распределения) в вероятностное распределение исходного набора данных. Генератор при обучении максимизирует вероятность того, что дискриминатор совершит ошибку. В пространстве функций G и D есть единственное решение, в рамках которого G восстанавливает распределение исходного набора обучающих данных, а значение D везде равно 1. Если обозначить образец из исходного набора данных через x , а шум на входе G через z , то дифференцируемая функция D должна быть такой, чтобы на исходных данных $D(x)$ принимала значение, близкое к 1. В этих обозначениях суть обучения модели кратко описывается следующим образом: в процессе обучения дифференцируемая функция D стремится стать такой, чтобы значение $D(G(z))$ стало близко к 0, и одновременно дифференцируемая функция G стремится стать такой, чтобы значение $D(G(z))$ стало близко к 1. Иными словами, D обучается максимизировать вероятность присвоения правильной метки как обучающим примерам, так и сгенерированным образцам, и одновременно G обучается минимизировать логарифмическую вероятность правильности дискриминатора $\log(1 - D(G(z)))$. В итоге такой минимаксной антагонистической игры достигается точка равновесия, которая является седловой точкой функции потерь дискриминатора $V(D, G)$

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(x)} [\log(1 - D(G(z)))].$$

Существенно, что ГСС в традиционных нейронных сетях обучается с помощью метода обратного распространения ошибки: в процессе оптимизации происходит обновление весов дискриминатора в сторону возрастания и весов генератора в сторону убывания соответствующих стохастических градиентов. В итоге обучения ГСС получает способность на основе случайного шума создавать уникальные изображения, имитирующие распределение признаков исходного набора данных.

2. МЕТОДЫ

В нашем исследовании мы сделали попытку реализации генеративно-состязательной сети на основе ИмНС с помощью пакета SNN Torch [20]. SNN Torch позволяет проектировать ИмНС, кодирующие информацию в импульсы как частотным, так и временным способом. При кодировании частотным методом, появление импульса (кодирующегося единицей) на определенном шаге можно рассматривать, как испытание Бернулли с одной попыткой. Преобразование из цифрового (например, интенсивность пикселя) в импульсный сигнал происходит следующим образом: нормализовав значение пикселя изображения от 0 до 1, мы получаем вероятность появления импульса на определенном шаге. Например, черный цвет пикселя будет иметь вероятность появления импульса равную 0, серый – равную 0.5, а белый будет иметь вероятность 1. Временной способ позволяет создавать более энергоэффективную модель, так как каждый нейрон способен производить лишь один импульс на протяжении всего временного ряда. На примере изображения появление импульса происходит следующим образом, чем ярче пиксель, чем дальше по временному ряду будет находиться импульс. Например, если временной ряд состоит из десяти шагов, то, чтобы закодировать абсолютно белый оттенок, импульс появится на десятом шаге, а для передачи абсолютно черного, импульс появится на первом. В нашей модели мы использовали частотное кодирование информации. Временное кодирование было оставлено для будущих исследований. Нейроны, составляющие ИмНС, моделировались интегрирующими нейронами (integrate-and-fire), где активация импульса (спайка) S в момент времени T происходит по следующей формуле [20]:

$$U[T] > U_{threshold} \implies S[T + 1] = 1,$$

а угасание потенциала мембраны определяется следующим выражением:

$$U[T + 1] = \beta U[T] + I_{in}[T + 1] - RU_{threshold}.$$

Здесь $U[i]$ – потенциал мембраны на шаге i , $U_{threshold}$ – пороговое значение, при достижении которого нейрон вырабатывает импульс, R – механизм сброса потенциала и β – коэффициент угасания потенциала мембраны. Так как спайковые нейронные сети не дифференцируемы, появляется проблема угасающих нейронов. В нашем случае она решалась с помощью суррогатного градиента для кусочно-постоянной функции активации. Мы использовали сглаживание fast sigmoid, где градиент описывается следующей формулой:

$$S \approx \frac{U}{1 + k|U|},$$

$$\frac{\partial S}{\partial U} = \frac{1}{(1 + k|U|)^2}.$$

На данный момент нами реализована рабочая схема ГСС, изображенная на Рис. 1.

На вход генератора (G) подается набор из $z = 128$ векторов нормального распределения. Затем сигнал циклом n -шагов проходит через спайковый генератор и на выходе мы получаем разряженный тензор спайков размером $(n, z, 28, 28)$. Параметры архитектуры сети представлены в Таблице 1. Здесь стоит отметить, что реализация SNN Torch позволяет комбинировать слои импульсной активации, со слоями пакета PyTorch, тем самым пользоваться уже написанными функциями библиотеки. В Таблице 1 в качестве деконволюционных слоев использовался реализованный класс обратной свертки из пакета PyTorch – ConvTranspose2D. Слои импульсной активации LIF (Leaky integrate-and-fire) нейрона и принцип его работы описан выше. Финальным слоем G выступает функция mean из пакета PyTorch, с помощью которой мы рассчитываем среднее значение по пространству n , тем самым получая изображение с минимальным значением яркости пикселя ноль и максимальным единицей.

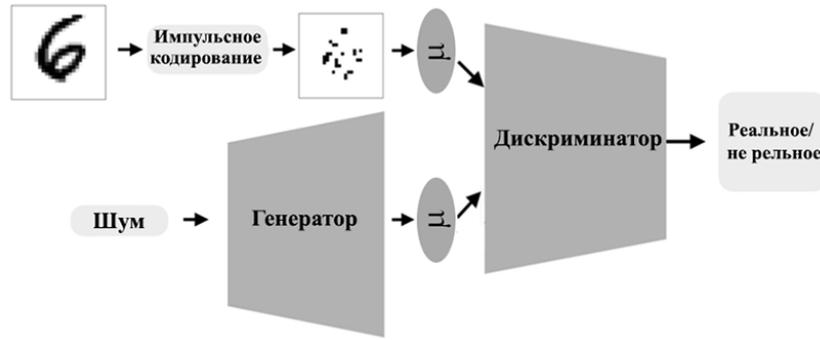


Рис. 1. Схема импульсной генеративной сети.

Таблица 1. Структура слоев импульсного генератора

Operation Layer	In Channels	Out Channels	Kernel Size	Stride	Padding
ConvTranspose	128	128	4	1	0
LIF					
ConvTranspose	128	64	3	2	1
LIF					
ConvTranspose	64	32	4	2	1
LIF					
ConvTranspose	32	1	4	2	1
Mean					

Получившиеся изображения мы подаем в аналоговый (традиционный) дискриминатор D , схема которого предствалена в Таблице 2.

Таблица 2. Структура слоев аналогового дискриминатора

Operation Layer	In Channels	Out Channels	Kernel Size	Stride	Padding
Conv	1	32	4	2	1
LeakyRelu					
Conv	32	64	4	2	1
LeakyRelu					
Conv	64	128	3	2	1
LeakyRelu					
Conv	128	1	4	1	0
Sigmoid					

Далее процесс обучения является стандартным для генеративно-сопоставительной сети. В качестве набора данных использовался популярный набор рукописных цифр MNIST. Сети G и D тренируются попеременно. В дискриминатор подаются реальные изображения MNIST и изображения, полученные с помощью генератора. Результаты оценки дискриминатора для каждого набора данных подаются в функцию потерь и происходит обновление весов дискриминатора. Для обучения генератора используются предсказанные метки дискриминатора, и по результатам функции потерь обновляются веса только генератора. В качестве функции потерь была выбрана бинарная кросс-энтропия. Для оптимизации градиентного спуска был выбран оптимизатор AdamW со следующими параметрами: скорость обучения 10^{-5} , коэффициент для вычисления скользящего среднего для градиента от 0.9 до 0.999 и коэффициент угасания 10^{-3} . Для каждой из сетей (G и D) инициализирован собственный оптимизатор. Для импульсных нейронов использовались следующие гиперпараметры. Количество шагов для генерации спайка $t = 30$, пороговое значение потенциала $U_{threshold} = 0.4$, коэффициент угасания

мембраны $\beta - 1$. Для суррогатного градиента, коэффициент $k - 25$.

В качестве метрики для оценки способности нейросети генерировать изображения была выбрана метрика inception score (IS) [21]. Для каждого сгенерированного изображения x вычисляется вектор вероятности принадлежности к каждому классу предобученной классификационной сети $p(x|y)$ где y это класс предобученной сети. В нашем случае мы использовали сеть-классификатор Inception v3. Затем мы рассчитываем расстояние Кульбака-Лейблера между распределением вероятности класса и общим распределением классов. Конечной формулой IS является экспонента математического ожидания полученных расстояний.

$$IS(G) = \exp(\mathbb{E}_{x \sim p_g}(D_{KL} p(y|x) || p(y))).$$

3. РЕЗУЛЬТАТЫ

На Рис. 2, 3, 4 показаны примеры изображений, синтезированные на выходе импульсного генератора на разных стадиях обучения. Как видно из Рис. 3, уже на семьдесят пятой эпохе изображения частично напоминают реальные цифры. При дальнейшем обучении визуальное качество изображений приближается к реальным изображениям.



Рис. 2. Сгенерированные изображения. Эпоха 0

Каждую эпоху мы записывали inception score для 1000 сгенерированных изображений. Увеличение inception score в процессе обучения отражено на Рис. 5. Высокие значения метрики указывают на то, что сгенерированная выборка изображений отличается резкостью и отчетливостью. Поведение функции потерь импульсного генератора и традиционного (аналогового) генератора приведены на Рис. 6. Для сравнения была натренирована аналоговая сверточная генеративно-состязательная сеть. Inception score аналогового GAN отражены на Рис. 7. Конструкция аналогового генератора была аналогична конструкции импульсного лишь с изменениями в слоях активации. Все импульсные активационные слои были заменены на связку BatchNorm и ReLU активацию. Сравнение графиков IS двух типов генераторов указывает на то, что скорость обучения и качество генерации находятся на схожем уровне.

4. ЗАКЛЮЧЕНИЕ

В данной работе мы разработали и продемонстрировали действующую модель спайковой генеративно-состязательной сети. Реализованная импульсная ГСС способна воспроизводить



Рис. 3. Сгенерированные изображения. Эпоха 75

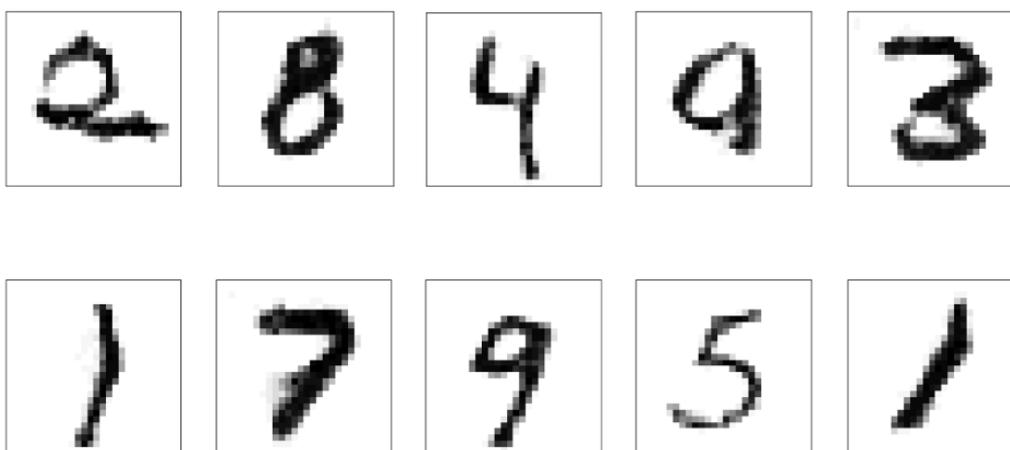


Рис. 4. Сгенерированные изображения. Эпоха 150

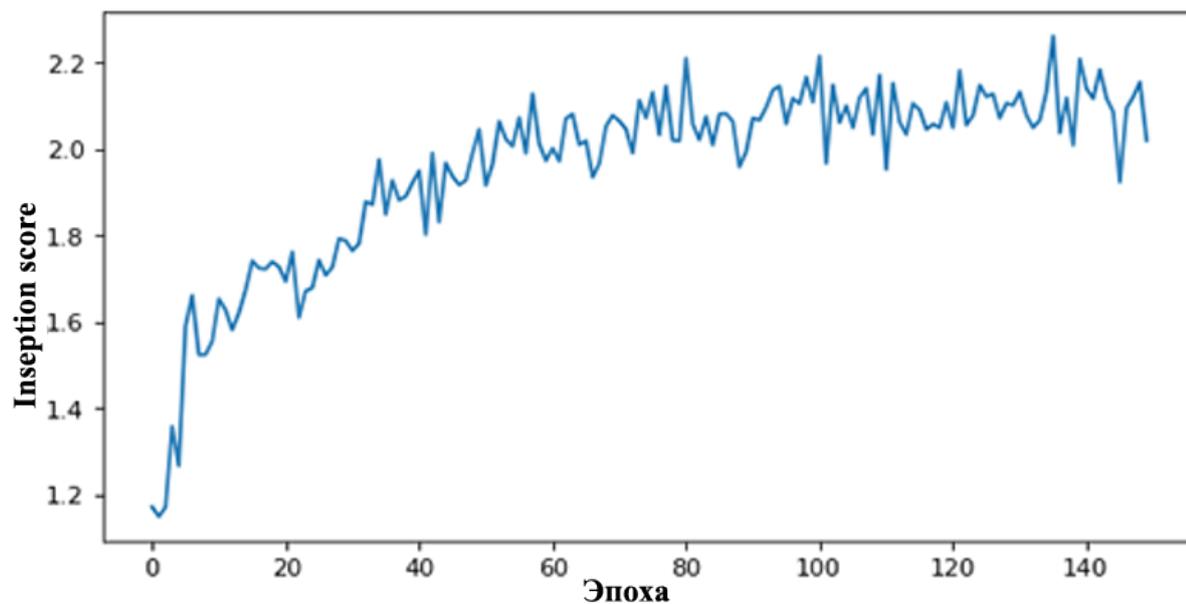


Рис. 5. Inception score. Импульсная генеративная сеть

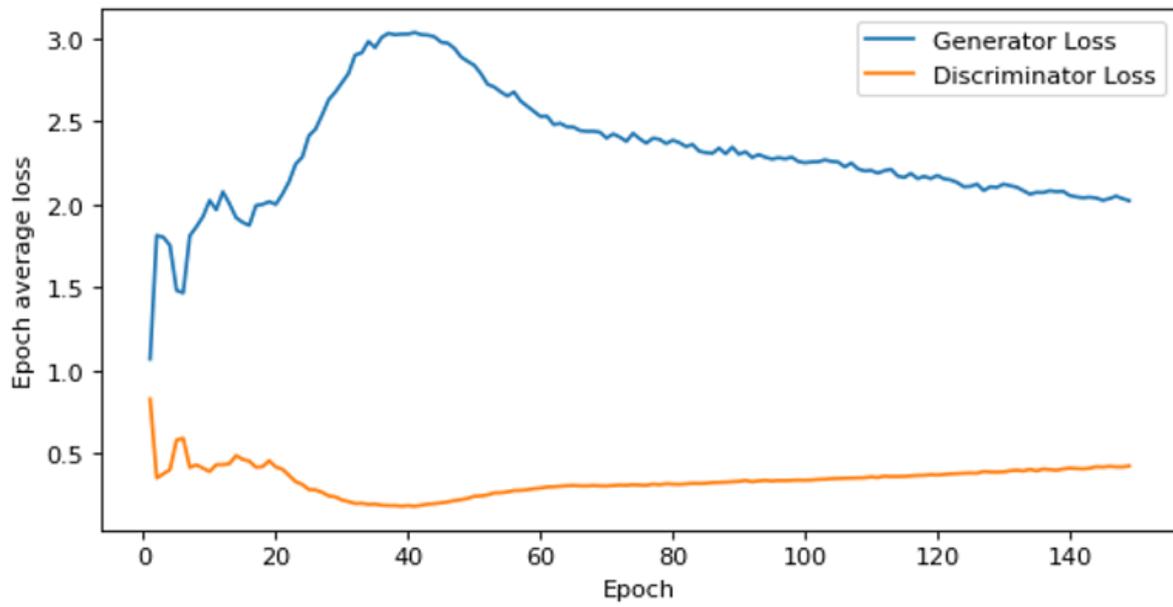


Рис. 6. Функции потерь дискриминатора и генератора ИмНС

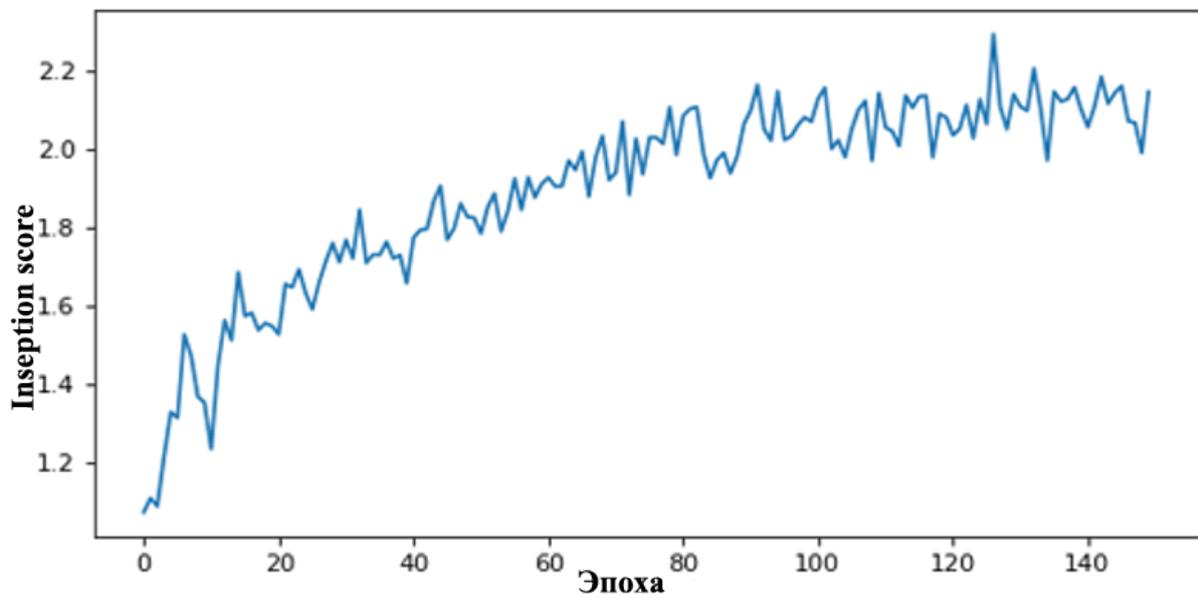


Рис. 7. Inception score. Аналоговая генеративная сеть

рукописный текст набора MNIST. Inception score синтезированных изображений импульсного ГСС находится примерно на том же уровне, что и inception score полностью аналогового ГСС. Inception Score набирает свое максимальное значение к сотой эпохе обучения.

В дальнейшем можно продолжать исследования как в сторону изменения типов сигнала, так и в создании других импульсных генеративных моделей, например условных генеративно-состязательных сетей.

СПИСОК ЛИТЕРАТУРЫ

1. Pfeiffer M., Pfeil T. Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in neuroscience*, 2018, no. 12, p.774.
2. Rueckauer B., Lungu I.A., Hu Y., Pfeiffer M. Theory and tools for the conversion of analog to spiking convolutional neural networks. *arXiv preprint*, 2016, arXiv:1612.04052.
3. Alqahtani H., Kavakli-Thorne M., Kumar G. Applications of generative adversarial networks (gans): An updated review. *Archives of Computational Methods in Engineering*, 2021, vol. 28, no. 2, pp. 525-552.
4. Maass W., Bishop C. M. *Pulsed neural networks*, MIT press, 2001.
5. Maass W. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 1997, vol. 10(9), pp. 1659-1671.
6. Ghosh-Dastidar S., Adeli H. Spiking neural networks. *International journal of neural systems*, 2009, no. 19(04), pp.295-308.
7. Taherkhani A., Belatreche A., Li Y., Cosma G., Maguire L.P., McGinnity T.M.. A review of learning in biologically plausible spiking neural networks. *Neural Networks*, 2020, 122, pp.253-272.
8. Izhikevich E. M. Simple model of spiking neurons. *IEEE Trans. Neural Networks*, 2003, vol. 14, pp, 1569-1572.
9. Izhikevich E. M. Which model to use for cortical spiking neurons? *IEEE Trans. Neural Networks*, 2004, vol. 15, pp. 1063-1070.
10. Paugam-Moisy H., Bohte S. Computing with Spiking Neuron Networks BT. *Handbook of Natural Computing* in, eds. G. Rozenberg, T. Back, and J. N. Kok, 2012, pp. 335-376.
11. Camunas-Mesa L. A., Linares-Barranco B., Serrano-Gotarredona T. Neuromorphic spiking neural networks and their memristor-CMOS hardware implementations. *Materials*, 2019, no. 12(17), p.2745.
12. Tang G., Shah A., Michmizos K. P. Spiking neural network on neuromorphic hardware for energy-efficient unidimensional slam. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 4176-4181.
13. Tavanaei A., Ghodrati M., Kheradpisheh S.R., Masquelier T., Maida A. Deep learning in spiking neural networks. *Neural Networks*, 2019, vol. 111, pp. 47-63.
14. Kotariya V., Ganguly U. Spiking-gan: A spiking generative adversarial network using time-to-first-spike coding. *arXiv preprint*, 2021.
15. Rosenfeld B., Simeone O., Rajendran B. Spiking generative adversarial networks with a neural network discriminator: local training, bayesian models, and continual meta-learning. *IEEE Transactions on Computers*, 2022, vol. 71, no. 11, pp. 2778-2791.
16. Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. Generative adversarial nets. *Proceedings of the 27th International Conference on Neural Information Processing Systems*. Cambridge: MIT Press, 2014, vol. 2, pp. 2672-2680.
17. Gui J., Sun Z., Wen Y., Tao D., Ye J. A review on generative adversarial networks: algorithms, theory, and applications. *arXiv preprint*, 2020, arXiv:2001.06937.
18. Saxena D., Cao J. Generative adversarial networks (gans) challenges, solutions, and future directions. *ACM Computing Surveys (CSUR)*, 2021, vol. 54, no. 3, pp. 1-42.

19. Мандрикова Б. С. Метод анализа данных сложной структуры с элементами машинного обучения *Компьютерная оптика*, 2022, т. 46. № 3. стр. 506-512.
20. Eshraghian J. K., Ward M., Neftci E., Wang X., Lenz G., Dwivedi G., Lu W. D. Training spiking neural networks using lessons from deep learning. *arXiv preprint*, 2021, arXiv:2109.12894.
21. Barratt S., Sharma R. A note on the inception score. *arXiv preprint*, 2018, arXiv:1801.01973.

Spiking Generative Adversarial Neural Networks with Frequency Encoding

Agliukov I. N., Antonov D.I., Sukhov S.V

Generative methods in artificial neural networks are currently experiencing explosive development. Compared to traditional artificial neural networks, Spiking Neural Networks (SNNs) offer greater energy efficiency when solving machine learning tasks. However, the training methods for SNNs are not fully developed, and generative methods for SNNs are virtually absent. This work proposes a generative adversarial network that encodes information with spikes. The network uses integrate-and-fire neurons. Information is encoded in frequency mode. The SNN was trained using a method approximating backpropagation of error. The generative network is implemented in the SNN Torch package. Experiments conducted on the publicly available MNIST dataset showed the ability of the generative SNN to create high-quality images.

KEYWORDS: spiking neural network, generative neural network, generative adversarial network.