Импульсные нейросети в генеративных flow matching моделях.

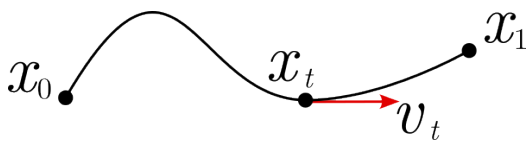
М.А. Прибыль, Н.В. Шмырёв

АЦ Технологии
contact@alphacephei.com

Импульсные нейронные сети (SNN) привлекают значительное внимание благодаря своей способности работать на нейроморфных устройствах со сверхвысокой скоростью и впечатляющей энергоэффективностью [2], [18]. В отличии от искусственных нейронных сетей, в которых информация имеет непрерывное представление, в SNN информация представлена в виде бинарных импульсов. Поэтому импульсные нейронные сети имитируют нейронную динамику мозга. Благодаря своим особенностям SNN широко применяются в различных задачах, включая распознавание [25], [4], отслеживание [24], обнаружение [10], [22], сегментацию [11] и восстановление изображений [6].

Диффузионные вероятностные модели шумоподавления [7], [19] или просто диффузионные вероятностные модели – это класс генеративных моделей, которые на каждом временном шаге добавляют гауссовый шум к данным (процесс прямой диффузии), а затем обучаются обращать процесс диффузии, чтобы получить исходные данные (процесс обратной диффузии). В последнее время диффузионные вероятностные модели приобрели популярность благодаря исключительным генеративным возможностям в таких областях, как генерация картинок ([7], [8]), видео [9], аудио данных ([12], [15]). Для достижения высокоточных результатов им требуется много итераций в процессе шумоподавления, что приводит к высоким вычислительным затратам. В сравнении с генеративно-состязательными нейросетями, где для генерации одного сэмпла требуется один проход через сеть, процесс генерации диффузионной вероятностной модели может состоять из тысячи шагов. Чтобы повысить скорость сэмплирования было предложено несколько моделей, наиболее популярные из которых неявная диффузионная модель шумоподавления (Denoising Diffusion Implicit Models) [20], скрытая диффузионная модель (Latent Diffusion Model) [17], а также модель согласования потоков (Flow Matching Model) [13]. Импульсные нейронные сети также являются перспективными кандидатами для ускорения процесса генерации диффузионных вероятностных моделей за счет использования своих возможностей для высокоскоростных вычислений с низким энергопотреблением. Однако исследования в данной области ограниченны. Насколько нам известно, существует всего четыре работы [1], [5], [16], [23]. В первых двух работах строятся импульсные диффузионные вероятностные модели для генерации картинок. Краткое описание этих работ можно найти в нашей предыдущей лекции `https://alphacephei.com/ru/lecture14.pdf`. В работе [16] мы исследовали импульсную диффузионную вероятностную модель, предложенную в [1], для генерации звуковых команд. Однако скорость генерации была недостаточной для более сложных задач. Чтобы ускорить процесс сэмплирования, в настоящей работе мы строим импульсный аналог генеративной модели Flow Matching для синтеза речи. В [23] также предлагалась модель, ускоряющая процесс сэмплирования импульсной диффузионной вероятностной модели. Авторы использовали Denoising Diffusion Implicit модель для построения импульсного аналога. Эту модель можно рассматривать, как частный случай Flow Matching при определенном выборе точечного потока и специальной связи между распределениями. Подробнее можно прочитать в [14].

Flow Matching или потоковое совпадение – это метод, который основан на той же идеи, что и диффузионные вероятностные модели, а именно постепенного ухудшения данных с помощью шума, а затем синтезирования новых данных с помощью обращения процесса. Сначала определим, что такое поток. Поток – это набор векторных полей, индексированных по времени $v = \{v_t\}_{t\in[0,1]}$. Любой поток определяет траекторию, переводящую начальную точку x_0 вдоль поля скоростей $\{v_t\}$ в конечную точку x_1.



Это можно записать в виде дифференциального уравнения:

$$\frac{dx_t}{dt} = v_t(x_t) \tag{1}$$

с граничными условиями x_0 и x_1 в момент $t = 0$ и $t = 1$ соответственно.

Простейшая конструкция потока для уравнения (1) с граничными условиями x_0 и x_1 – это

$$v_t(x_t) = x_1 - x_0. \tag{2}$$

Тогда решением обыкновенного дифференциального уравнения (1) с соответствующими граничными условиями будет линейная интерполяция между x_0 и x_1:

$$x_t = (1-t)x_0 + tx_1, \quad t \in [0,1].$$

Пободно тому, как поток определяет отображение между начальной и конечной точками, он также определяет отображение между распределениями. Основная цель Flow matching определить поток v^*, который переносит вероятностное распределение начальных точек p_0 в распределение конечных точек p_1, т.е.

$$p_0 \xhookrightarrow{v^*} p_1.$$

Распределение p_0 – это обычно Гауссово распределение, а p_1 – заданное распределение данных. Для того, чтобы определить поток v^*, строися нейронная модель $f_\theta(x_t, t)$ с параметрами θ, минимизирующая функцию потерь:

$$L = \mathbb{E}_{(x_0,x_1,x_t)}\|v_t - f_\theta(x_t,t)\|^2 \to \min, \tag{3}$$

где v_t – векторное поле из уравнения (1), которое перемещает точку x_0 в x_1. Чтобы сгенерировать новый сэмпл, мы строим генеративную модель

$$\frac{dx_t}{dt} = f_\theta(x_t,t),\ x_0 \sim p_0,\ x_1 \sim p_1,$$

из которой методом Эйлера находим правило обновления

$$x_{t+\Delta t} = x_t + \Delta t f_\theta(x_t,t)\ \ \forall t \in [0,1). \tag{4}$$

Если сравнить прямые процессы зашумления данных диффузионных вероятностных моделей и Flow Matching, то они похожи, так как являются линейной интерполяцией данных и шума. А вот обратные процессы, то есть очищения данных от шума, отличаются. В обратном процессе диффузии присутствует шумовая составляющая, которая является причиной "зубчатых" траекторий. В обратном процессе Flow Matching (4) шумовой составляющей нет, поэтому траектории гладкие. Гладкость или прямизна траектории имеет решающие значение, поскольку определяет количество шагов, необходимых для достижения точных результатов, обеспечивая компромисс между вычислительными затратами и точностью. Более прямые траектории требуют меньшее количество шагов для генерации, тогда как траектории с большей кривизной требуют больше шагов. Следовательно, модели Flow Matching открывают возможности для повышения скорости и производительности вывода.

Система синтеза речи Vosk-TTS имеет архитектуру, основанную на подходе Latent Flow Matching и диффузионных трансформерах DiT. Latent Flow Matching представляет собой модификацию Flow Matching, где преобразования происходят в скрытом пространстве, а не в пространстве данных. Это позволяет более эффективно обучать модель и уменьшить вычислительные затраты, так как размерность скрытого пространства обычно меньше, чем размерность пространства исходных данных.

Процесс синтеза начинается с текста, который проходит через Encoder, а затем вместе с шумом подается на основную часть архитектуры – Flow Matching для точной генерации спектрограмм Mel. Полученные спектрограммы преобразуются в звуковые волны с помощью декодера Vocos, который обеспечивает высокое качество синтеза речи.

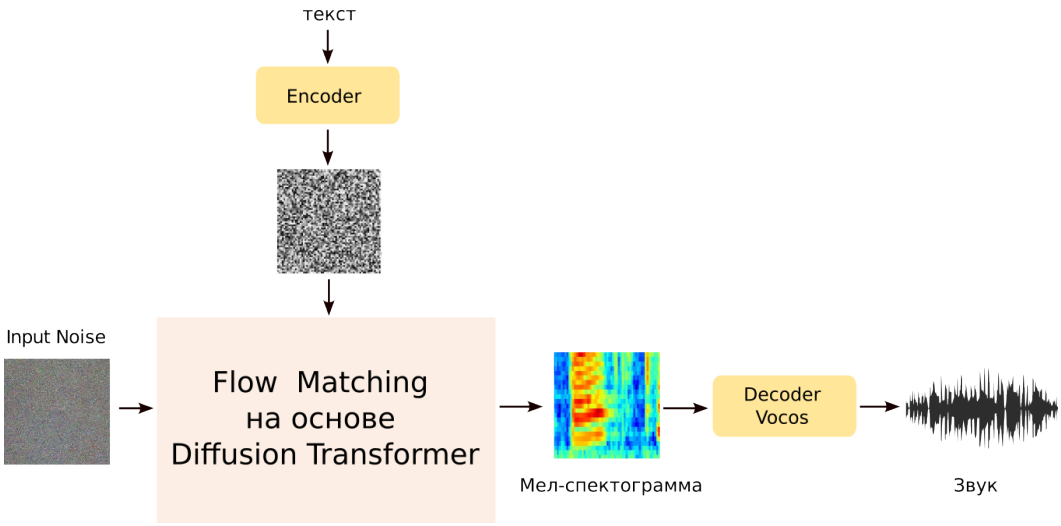


Рис. 1: Архитектура синтеза речи Vosk-TTS .

Для построения импульсного аналога синтеза речи Vosk-TTS мы преобразовываем только основную часть архитектуры – Flow Matching, хотя уже существует работа, в которой декодер Vocos также переведен в импульсный аналог [3]. Преобразование диффузионных трансформеров, на которых основана модель Flow Matching, является трудновыполнимой задачей из-за их сложной архитектуры. Поэтому мы рассматриваем Flow Matching на основе свёрточной U-net. И хотя свёрточная U-net даёт немного хуже качество синтеза, тем не менее она оказывается более подходящей для преобразования в импульсный аналог. Кроме того, в работе [16] мы уже использовали импульсную свёрточную U-net. Архитектура импульсной модели Flow Matching представлена ниже.

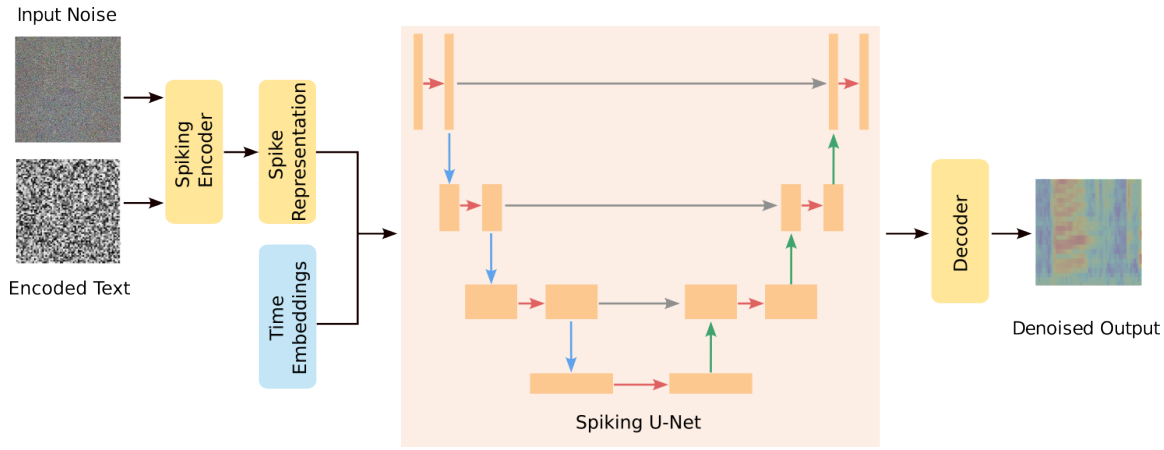


Рис. 2: Flow Matching модель на основе импульсных нейронных сетей.

Гауссовый шум вместе с закодированным текстом преобразуется энкодером в импульсную последовательность и вместе с Time Embedding последовательно подается в импульсную нейронную сеть U-net. Time embeddings – это векторы, которые отмечают каждую 32 миллисекунду или каждый отсчет, чтобы нейросеть могла их отличать. Импульсная U-net передает только импульсы. Она состоит из сужающегося пути и расширяющегося пути. На каждом этапе сужающегося пути разрешение данных уменьшается, а количество каналов свойств увеличивается. Каждый шаг расширяющегося пути, напротив, восстанавливает разрешение и уменьшает количество каналов. Эти два пути связаны пропускными соединениями, которые необходимы для восстановления пространственной информации, потерянной при уменьшении разрешения, а также для решения проблемы недостаточно точного восстановления границ из низкоразмерного представления. Далее, выходящие импульсы из U-net передаются в декодирующий слой, после чего вычисляется шумоподавляющий шаг.

После фиксированного числа шумоподавляющих шагов мы получаем новый сэмпл.

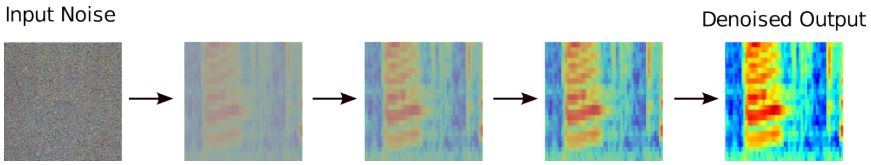


Рис. 3: Сэмплирование.

Тренировка импульсной модели Flow Matching проводится без энкодера. Для тренировки используется специальный пакет Python для обучения импульсных нейронных сетей SpikingJelly [21], в котором алгоритм обучения основан на методе суррогатного градиента. Однако, при приближении значений t к границам 0 и 1 градиент функции потерь становится очень большим. На начальном этапе генерации это приводит к нестабильности тренировки, а на последних этапах генерации, когда шум практически исчезает, осложняет корректировку сигнала. Поэтому при $t < 0.01$ пакет данных для тренировки мы берем не часто, а верхнюю границу $t = 1$ уменьшаем до 0.98. Более того, тренировка проводится с учителем, то есть с аналогичной не импульсной моделью Flow Matching. Такой метод тренировки был предложен в работе [3].

Для тренировки используется база **Biggest RU Book Cleanup**, которая состоит 200 дикторов, 2000 часов записи речи (`https://huggingface.co/datasets/alphacep/biggest-ru-book-cleanup`). Результаты отражены в следующей таблице.

Таблица 1: Результаты

Model	FAD ↓	WER ↓	SS ↑
Vosk-TTS	0.96	0.9	0.76
импульсный Vosk-TTS	1.25	2.0	0.55

В качестве метрики мы использовали:

1. Frechet Audio Distance (FAD): измеряет расстояние между вероятностными распределениями реальных и синтезированных аудиофайлов (характеристика интонаций);

2. Word Error Rate (WER): измеряет точность распознавания слов в синтезированном аудиофайле по сравнению с эталонным текстом (позволяет оценить наличие неправильно произнесенных слов);

3. Speaker Similarity (SS): измеряет сходство синтезированного голоса с голосом оригинального диктора. Высокие значения SS говорят о том, что синтезированная речь очень похожа на речь конкретного человека.

В результате исследования мы убедились, что преобразование генеративной модели Flow Matching на основе свёрточной U-net в импульсный аналог дает неплохие результаты, и может применяться при наличии доступной аппаратуры. Сэмплирования по сравнению с импульсной диффузионной вероятностной моделью для генерации речевых команд [16] существенно ускоряется. Если импульсной диффузионной вероятностной модели необходимо более 100 шагов для генерации речевой команды, то flow matching модели всего 10. Не смотря на то, что тренировка модели осложняется нестабильностью в граничных точках, тем не менее благодаря этому увеличивается разнообразие генерируемых данных. Без нестабильности генерируемые данные были бы похожи.

Данная работа была представлена на конференции `https://neuro.kaspersky.ru/conference/`. Код доступен по ссылке `https://github.com/alphacep/SDDPM`.

Литература

[1] J. Cao, Z. Wang, H. Guo, H. Cheng, Q. Zhang, R. Xu. Spiking denoising diffusion probabilistic models, arXiv preprint arXiv:2306.17046, 2023.

[2] A. S. Cassidy, R. Alvarez-Icaza, F. Akopyan, J. Sawada, J. V. Arthur, P. A. Merolla, P. Datta, M. G. Tallada, B. Taba, A. Andreopoulos, et al. Real-time scalable cortical computing at 46 gigasynaptic ops/watt with $\sim 100\times$

speedup in time-to-solution and $\sim 100,000\times$ reduction in energy-to-solution. In SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2014.

[3] Yu. Chen, Zh. Mu, A. Li, P. Li, X. Yang. Spiking vocos: an energy-efficient neural vocoder. arXiv preprint arXiv:2509.13049v1, 2025.

[4] S. Deng, Y. Li, S. Zhang, and S. Gu. Temporal efficient training of spiking neural network via gradient reweighting, arXiv preprint arXiv:2202.11946, 2022.

[5] M. Liu, J. Gan, R. Wen, T. Li, Y. Chen, H. Chen. Spiking-Diffusion: vector quantized discrete diffusion model with spiking neural networks, arXiv preprint arXiv:2308.10187v4, 2023.

[6] S. Li, Y. Feng, Y. Li, Y. Jiang, C. Zou, and Y. Gao. Event stream super-resolution via spatiotemporal constraint learning, in ICCV, 2021.

[7] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

[8] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans. Cascaded diffusion models for high fidelity image generation, J. Mach. Learn. Res., vol. 23, no. 47, 2022.

[9] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models, arXiv preprint arXiv:2204.03458, 2022.

[10] S. Kim, S. Park, B. Na, and S. Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection, in AAAI, vol. 34, no. 07, 2020.

[11] P. Kirkland, G. Di Caterina, J. Soraghan, and G. Matich. Spikeseg: Spiking segmentation via stdp saliency mapping, in IJCNN. IEEE, 2020.

[12] Zh. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. Diffwave: A versatile diffusion model for audio synthesis, International Conference on Learning Representations, 2021.

[13] Y. Lipman, R. TQ Chen, H. Ben-Hamu, M. Nickel, and Matt Le. Flow matching for generative modeling, arXiv preprint arXiv:2210.02747, 2022.

[14] P. Nakkiran, A. Bradley, H. Zhou, M. Advani. Step-by-Step Diffusion: An Elementary Tutorial, arXiv preprint arXiv:2406.08929v2, 2024.

[15] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov. Gradtts: A diffusion probabilistic model for text-to-speech, International Conference on Machine Learning. PMLR, 2021.

[16] М.А. Прибыль, Н.В. Шмырёв. Импульсная диффузионная модель генерации речевых команд. `https://alphacephei.com/ru/lecture13.pdf`

[17] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.

[18] K. Roy, A. Jaiswal, and Priyadarshini Panda. Towards spike-based machine intelligence with neuromorphic computing. Nature, 575(7784), 2019.

[19] Y. Song, J. Sohl-Dickstein, D. P Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations, arXiv preprint arXiv:2011.13456, 2020.

[20] J. Song, Ch. Meng, S. Ermon. Denoising diffusion implicit models, arXiv preprint arXiv:2010.02502v4, 2022.

[21] SpikingJelly, `https://spikingjelly.readthedocs.io/zh-cn/latest/index.html#index-en` .

[22] M. Yuan, C. Zhang, Z. Wang, H. Liu, G. Pan, and H. Tang. Trainable spiking-yolo for low-latency and high-performance object detection, Neural Networks, vol. 172, 2024.

[23] R. Watanabe, Y. Mukuta, T. Harada. Fully spiking denoising diffusion implicit models, arXiv preprint arXiv:2312.01742v1, 2023.

[24] J. Zhang, B. Dong, H. Zhang, J. Ding, F. Heide, B. Yin, and X. Yang. Spiking transformers for event-based single object tracking, in CVPR, 2022.

[25] Z. Zhou, Y. Zhu, C. He, Y. Wang, S. Yan, Y. Tian, and L. Yuan. Spikformer: When spiking neural network meets transformer, arXiv preprint arXiv:2209.15425, 2022.