

ГЕНЕРАЦИЯ ВРЕМЕННЫХ СИГНАЛОВ ИЗ СТАТИЧЕСКИХ ИЗОБРАЖЕНИЙ ДЛЯ ПОДАЧИ НА СПАЙКОВЫЕ НЕЙРОННЫЕ СЕТИ

А. С. Тощев^[0000-0003-4424-6822]

Казанский (Приволжский) федеральный университет, г. Казань, Россия

atoschev@kpfu.ru

Аннотация

Спайковые нейронные сети (далее — СНС, т. е. нейросети, передающие информацию во времени с помощью импульсов) требуют временного входа, тогда как в задачах компьютерного зрения данные чаще заданы статическими изображениями. В работе рассмотрено преобразование вида «изображение – временной сигнал – импульсы» и исследовано влияние способа входного кодирования на динамику обучения СНС, плотность импульсной активности и вычислительную стоимость обработки. В экспериментальной части реализованы и сопоставлены два семейства кодирования: кодирование по времени первого импульса (Latency) и пуассоновское кодирование по интенсивности (Poisson); для них рассмотрены четыре режима: базовый Latency без подавления фона, модифицированный Latency с порогом тишины, стохастический Poisson и детерминированный Poisson. В качестве метрик использованы среднее число импульсов на пример, число синаптических операций, прокси-показатель энергозатрат и характеристики конкуренции нейронов скрытого слоя. Эксперименты на наборе MNIST (60000 обучающих и 10000 тестовых изображений) для сети со скрытым слоем из 100 нейронов и горизонтом моделирования 200 шагов показали, что все исследованные режимы обеспечивают устойчивое обучение без коллапса активности. При этом модифицированный Latency с порогом тишины $x_{\min} = 0.05$ оказался наиболее эффективным по соотношению «полезная активность — вычислительная стоимость»: при количестве спайков на один пример 323.41 для него число синаптических операций составило 14295.09, тогда как базовый Latency без фильтрации фона при близкой выходной активности (311.22 импульса на пример) потребовал 78400 синаптических операций.

Ключевые слова: *спайковые нейронные сети, распознавание изображений, кодирование сигнала, кодирование изображений.*

ВВЕДЕНИЕ

Спайковые нейронные сети (СНС) представляют собой класс моделей, в которых вычисления и передача информации задаются во времени, а базовой единицей взаимодействия между нейронами выступает импульс — кратковременное дискретное событие. В отличие от традиционных нейронных сетей, работающих с непрерывными активациями, СНС опираются на временную структуру сигналов и редкие события, что рассматривается как одно из принципиальных отличий третьего поколения нейросетевых моделей [1, 2].

Такой принцип организации делает СНС естественно близкими к биологическим нейронным системам и одновременно открывает путь к событийно-управляемым вычислениям, в которых вычислительная стоимость определяется не столько числом арифметических операций, сколько количеством сгенерированных импульсов и длительностью их обработки. Именно поэтому СНС и нейроморфные системы рассматриваются как перспективное направление для энергоэффективных вычислений, особенно в условиях ограниченных ресурсов и на специализированном аппаратном обеспечении [3].

Однако применение СНС в задачах компьютерного зрения связано с принципиальной трудностью. В большинстве практических постановок входные данные представлены статическими изображениями, тогда как сами СНС наиболее естественно работают с временными или событийными потоками. Это означает, что пространственная структура изображения должна быть специально преобразована во временную последовательность импульсов, причем выбор способа такого преобразования существенно влияет как на качество распознавания, так и на задержку, число событий и вычислительную стоимость. В литературе именно схема кодирования входа рассматривается как один из ключевых факторов, определяющих свойства спайковой сети при обучении и распознавании [4, 5].

Постановка задачи, рассмотренной ниже, заключается в преобразовании статического изображения во временную последовательность импульсных со-

бытий, пригодную для обработки спайковой нейронной сетью. Такое преобразование должно одновременно сохранять информацию, существенную для распознавания, обеспечивать управляемую разреженность импульсной активности и допускать контроль задержки принятия решения. От выбора схемы кодирования в этом случае зависит не только информативность представления, но и режим работы сети в целом [5, 6].

Целью настоящего исследования были разработка и анализ воспроизводимой схемы преобразования изображения во временной вход для спайковых нейронных сетей, а также экспериментальное исследование влияния способа кодирования и его параметров на характеристики импульсной активности и вычислительную стоимость на наборе MNIST [7]. Предложен единый формализм «изображение – временной сигнал – импульсы», выполнено сопоставление реализованных схем кодирования входа, проведено количественное сравнение режимов входного кодирования по показателям импульсной активности, вычислительной стоимости и конкуренции нейронов скрытого слоя и сформирован воспроизводимый набор метрик для анализа таких систем с точки зрения активности, вычислительной стоимости и распределения конкуренции между нейронами [4, 6].

МЕТОДОЛОГИЯ

Были использованы нейроны типа «интегратор с утечкой и порогом» (распространенная модель спайкового нейрона) [8], а обучение в СНС выполнено по правилу пластичности, зависящей от относительного времени импульсов [9].

Требуется построить преобразование статического изображения в последовательность импульсов $X \rightarrow s(t) \rightarrow z(t)$, после чего подать $z(t)$ на СНС и решить задачу классификации. Нас интересует компромисс: точность при ограничениях на число импульсов и задержку [6].

Преобразование «изображение – временной сигнал»

Рассмотрены два семейства кодирования:

- 1) кодирование по интенсивности (частотное, Poisson);
- 2) кодирование по времени первого импульса (Latency) [4].

Кодирование по интенсивности (частотное, Poisson)

Пуассоновское кодирование относится к частотным способам представления входа: интенсивность пикселя интерпретируется как интенсивность генерации импульсов во времени. Такой подход широко используется в спайковых сетях для задач распознавания изображений, в том числе в классической постановке Diehl–Cook [4] для MNIST, где входные изображения подаются в виде пуассоновских спайковых последовательностях с частотой, пропорциональной яркости пикселя.

Пусть исходное изображение задано матрицей $X = \{x_{ij}\}$, где $x_{ij} \in [0,1]$. В нашей реализации после нормировки изображение преобразуется в вектор яркостей, а для каждого пикселя задается параметр интенсивности

$$\lambda_{ij} = x_{ij}\beta, \quad (1)$$

где $\beta > 0$ — коэффициент масштабирования интенсивности (*rate_scale*). Формула (1) задает рабочую точку энкодера: при малых β вход становится слишком разреженным, а при больших β резко растет плотность событий.

Далее вероятность появления импульса на одном шаге времени вычисляется как

$$p_{ij} = 1 - \exp(-\lambda_{ij}). \quad (2)$$

Формула (2) соответствует стандартной связи между интенсивностью пуассоновского процесса и вероятностью события на дискретном временном шаге. В практических реализациях СНС такой способ кодирования используется как базовый вариант частотного представления входа [4].

После этого для каждого шага времени $t = 1, \dots, T$ и каждого пикселя выполняется независимое испытание Бернулли:

$$z_{ij}(t) = \mathbb{I}(u_{ij}(t) < p_{ij}), \quad u_{ij}(t) \sim U(0,1), \quad (3)$$

где $z_{ij}(t) \in \{0,1\}$ — наличие импульса на шаге t . В результате получается бинарный импульсный поток длины T , который подается на вход спайковой сети.

Интерпретация параметра β

Параметр β в формулах (1) и (2) управляет компромиссом «информативность – число импульсов». При слишком малых значениях β вероятность p_{ij} близка к нулю, и сеть недополучает событий. При увеличении β растет число входных импульсов и, как правило, повышается информативность входа, однако одновременно возрастают вычислительная стоимость и риск насыщения активности. Сравнительные исследования схем кодирования в СНС также показывают, что частотное кодирование чувствительно к длительности окна и плотности событий, поскольку информация здесь извлекается из накопления импульсов во времени.

Воспроизводимый режим

Поскольку стандартное пуассоновское кодирование стохастично, повторное кодирование одного и того же изображения может давать разные импульсные последовательности. Это допустимо с точки зрения модели, но затрудняет сравнение энкодеров и построение воспроизводимых графиков. Поэтому в реализации предусмотрен детерминированный режим (`deterministic = True`), в котором псевдослучайная последовательность зависит от изображения и базового зерна генератора. Формально это можно записать в форме

$$u_{ij}(t) = f(X, \text{seed}, t, i, j), \quad (4)$$

где функция $f(\cdot)$ строит воспроизводимую псевдослучайную последовательность в интервале $[0,1)$. При этом правило сравнения (3) сохраняется, но одинаковое изображение кодируется одинаково при каждом повторном запуске.

Кодирование по времени первого импульса (Latency)

В этом семействе яркость кодируется моментом появления первого события: яркие пиксели должны давать импульс раньше [10]. Для каждого пикселя вычисляется время

$$t_{ij}^* = 1 + \lfloor (T - 1)(1 - x_{ij})^\gamma \rfloor, \quad \gamma > 0. \quad (5)$$

Далее считаем, что в момент t_{ij}^* возникает событие (либо формируется узкий всплеск сигнала вокруг этого момента).

Формула (5) обеспечивает упорядочивание по важности: пиксели с большей яркостью дают меньшие t_{ij}^* , т. е. информация поступает в сеть раньше. Параметр γ в (5) задает «контраст по времени»: при увеличении γ различия между яркими областями сильнее сжимаются к ранним шагам.

Кодирование по времени первого импульса задает для каждого пикселя ровно одно событие, а момент события определяется яркостью: чем ярче пиксель, тем раньше он «срабатывает». В исходной форме такой подход порождает событие для каждого пикселя, включая фон. Для изображений MNIST это приводит к нежелательному эффекту: фоновые пиксели с $x_{ij} \approx 0$ формируют массовый «залп» на последнем шаге времени $t = T - 1$, поскольку для них значение t_{ij}^* оказывается максимальным. Этот залп не несет полезной информации, но резко увеличивает плотность событий и может ухудшать стабильность обучения.

Чтобы устранить данный эффект, в кодировщик был введен порог тишины x_{\min} . Пиксели с яркостью ниже x_{\min} считаются фоновыми и не генерируют событие вообще. Таким образом, события формируются только по «значимым» пикселям изображения.

Формально, после нормировки яркости $x_{ij} \in [0,1]$, введем маску активности

$$m_{ij} = \mathbb{I}[x_{ij} \geq x_{\min}].$$

Для пикселей с $m_{ij} = 1$ вычислим момент первого события:

$$t_{ij}^* = \lfloor (1 - x_{ij})(T - 1) \rfloor \quad \text{только для } m_{ij} = 1.$$

Далее импульсный поток зададим в виде

$$z_{ij}(t) = \begin{cases} 1, & t = t_{ij}^* \text{ и } m_{ij} = 1, \\ 0, & \text{иначе} \end{cases}$$

В результате модификации достигаются два практических эффекта.

1. Устраняется «залп фона» на последнем шаге времени: если $x_{ij} < x_{\min}$, то событие не создается, и фон не формирует массовую активность в момент $t = T - 1$.

2. Снижается число входных событий и повышается информативность временной структуры: события концентрируются на пикселях штрихов цифры, а не на пустом фоне.

В проведенных оценках корректности кодировщика было показано, что для полностью нулевого изображения при $x_{\min} > 0$ поток событий отсутствует, а для разреженных изображений число событий равно числу пикселей, удовлетворяющих условию $x_{ij} \geq x_{\min}$. Это обеспечивает воспроизводимое и интерпретируемое кодирование для дальнейшего обучения и сравнения режимов по метрикам: среднее число импульсов, индекс концентрации победителей, количество задействованных нейронов.

Постановка экспериментов

Было выполнено сравнение четырех режимов кодирования входного изображения в импульсную последовательность:

- 1) **Latency, all pixels** — базовое кодирование по времени первого импульса без подавления фона (`latency_x_min = 0`);
- 2) **Latency, filtered** — модифицированное кодирование по времени первого импульса с порогом тишины (`latency_x_min = 0.05`);
- 3) **Poisson, stochastic** — стохастическое пуассоновское кодирование по интенсивности (`poisson_deterministic = False`);
- 4) **Poisson, deterministic** — детерминированный пуассоновский режим (`poisson_deterministic = True`).

Во всех случаях была использована одна и та же архитектура сети: входной слой размерности 784, скрытый слой из 100 нейронов типа «интегратор с утечкой и порогом», горизонт моделирования $T = 200$ шагов. Обучение выполнялось на 60000 изображений MNIST с одинаковыми базовыми гиперпараметрами STDP, инициализации весов, торможения и гомеостатической подстройки порогов. Таким образом, различие между экспериментами определялось только с помощью кодирования входного изображения. Для пуассоновских режимов были использованы `poisson_rate_scale = 0.011` и дополнительное усиление входа `encoder_rate_boost = 7`. Для латентностного режима с фильтрацией фона применялся порог тишины `latency_x_min = 0.05`, при котором пиксели с яркостью ниже этого значения не генерировали импульсов.

Оцениваемые характеристики

Для каждого режима кодирования анализировалась реакция сети в процессе обучения. Для этого фиксировались следующие метрики:

- среднее число импульсов на пример (spikes_per_sample);
- число синаптических операций на пример (synops_per_sample);
- интегральный прокси-показатель вычислительной стоимости (energy_proxy_per_sample);
- число нейронов, выступавших победителями хотя бы один раз (winners_unique);
- индекс концентрации победителей (winner_HHI).

Такой набор метрик позволяет оценивать не только способность сети обучаться, но и характер формирующейся специализации скрытого слоя: равномерность использования нейронов, степень конкуренции и вычислительную «стоимость» работы сети.

РЕЗУЛЬТАТЫ

Итоги обучения представлены в табл. 1.

Табл. 1. Результаты обучения сети при разных режимах кодирования входа

Режим кодирования	Spikes per sample	Synops per sample	Energy proxy per sample	Winners unique	Winner HHI
Latency, all pixels	311.22	78400.00	4331.22	100	0.01133
Latency, filtered	323.41	14295.09	1138.16	100	0.01119
Poisson, stochastic	451.31	22423.99	1672.51	100	0.01022
Poisson, deterministic	450.36	22426.79	1671.70	100	0.01022

Полученные данные показывают, что все исследованные режимы обеспечивают устойчивое обучение без коллапса активности: во всех случаях $\text{winners_unique} = 100$, т. е. весь скрытый слой участвует в работе, а значения winner_NNI близки к $1/100$, что соответствует почти равномерному распределению победителей по нейронам.

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ

Наиболее показательное различие наблюдается между двумя вариантами Latency-кодирования. Базовый режим Latency, all pixels генерирует событие для каждого пикселя, включая фон. Это приводит к крайне большому числу синаптических операций (78400.0 на пример) и высокому энергетическому прокси (4331.22), хотя полезная выходная активность скрытого слоя остается умеренной (311.22 импульса на пример).

Введение порога тишины в режиме Latency, filtered радикально изменяет ситуацию. При $\text{latency_x_min} = 0.05$ фоновая активность подавляется, что приводит к резкому снижению вычислительной стоимости: synops_per_sample уменьшается более чем в 5 раз, а $\text{energy_proxy_per_sample}$ — почти в 4 раза. При этом среднее число импульсов скрытого слоя не только не падает, но даже немного увеличивается (323.41 против 311.22). Это означает, что фильтрация фона практически не ухудшает полезную активность сети, но устраняет значительное число нефункциональных входных событий [4, 10].

Оба пуассоновских режима — стохастический и детерминированный — формируют более плотную активность скрытого слоя (≈ 450 импульсов на пример), чем модифицированный Latency, и имеют более высокую вычислительную стоимость, но остаются существенно более экономичными, чем Latency без фильтрации фона. При этом различия между стохастическим и детерминированным пуассоновским кодированием практически отсутствуют: значения spikes_per_sample , synops_per_sample , $\text{energy_proxy_per_sample}$, winners_unique и winner_NNI совпадают с высокой точностью. Следовательно, детерминированный пуассоновский режим может рассматриваться как воспроизводимая альтернатива стохастическому без заметного изменения характера обучения.

ОГРАНИЧЕНИЯ И БУДУЩАЯ РАБОТА

Несмотря на полученные устойчивые режимы обучения для различных способов кодирования входа, проведенное исследование имеет ряд ограничений.

Во-первых, сравнение выполнялось на относительно простой задаче распознавания рукописных цифр MNIST. Хотя этот набор данных удобен для анализа динамики обучения и свойств кодировщиков, он не отражает всей сложности реальных зрительных задач, где присутствуют текстуры, сложный фон, межклассовое сходство и более высокое пространственное разрешение. Поэтому сделанные выводы о преимуществах того или иного энкодера не следует автоматически переносить на более сложные наборы данных без дополнительной проверки.

Во-вторых, в работе исследована одна фиксированная архитектура сети: входной слой размерности 784, скрытый слой из 100 нейронов типа «интегратор с утечкой и порогом», один обучаемый полносвязный слой и заданный набор параметров STDP, торможения и гомеостатической подстройки порогов. Полученные результаты показали, что даже в такой конфигурации выбор кодировщика существенно влияет на плотность активности и вычислительную стоимость, однако влияние этих факторов может изменяться при увеличении числа нейронов, глубины сети или изменении структуры связей.

В-третьих, оценка на данном этапе сосредоточена преимущественно на динамике обучения сети, а не на окончательном качестве распознавания. В частности, основной акцент сделан на таких метриках, как число импульсов на пример, число синаптических операций, прокси-энергии и распределение победителей по нейронам. Эти метрики важны для понимания устойчивости и вычислительной эффективности сети, однако они не заменяют полноценного анализа итоговой точности классификации после этапа построения карты меток нейронов и последующего выходного классификатора.

В-четвертых, латентностный энкодер был модифицирован введением порога тишины x_{\min} , что позволило подавить нефункциональную активность фона. Хотя экспериментально показано, что такая модификация существенно снижает вычислительную стоимость без потери полезной активности, оптимальное значение x_{\min} в настоящей работе систематически не подбиралось. Аналогично,

для пуассоновского кодирования использовалось фиксированное усиление входа, тогда как более подробное исследование по сетке параметров могло бы дать более точные рекомендации по выбору рабочей точки.

Наконец, детерминированный пуассоновский режим рассматривался как воспроизводимая альтернатива стохастическому, однако не анализировалось, в какой степени его использование влияет на итоговую способность сети к обобщению по сравнению с полностью стохастическим режимом.

Направления будущей работы

Перспективы дальнейших исследований связаны как с расширением экспериментальной базы, так и с углублением анализа самих механизмов кодирования.

Во-первых, следующим шагом является полноценная оценка качества классификации для всех исследованных режимов кодирования после построения карты меток нейронов и применения выходных схем распознавания. Это позволит сопоставить не только вычислительную стоимость и структуру активности, но и итоговую точность классификации, а также компромисс «точность — число импульсов».

Во-вторых, представляет интерес систематический перебор параметров энкодеров. Для латентностного кодирования это прежде всего порог тишины x_{\min} , а также возможные нелинейные преобразования яркости во время первого импульса. Для пуассоновского кодирования важнейшими параметрами являются коэффициент масштабирования интенсивности и дополнительное усиление входа. Такой анализ позволит построить кривые зависимости качества и вычислительной стоимости от параметров кодирования и выбрать оптимальные рабочие режимы.

В-третьих, перспективно распространить предложенный подход на более сложные наборы данных и архитектуры, включая многослойные спайковые сети, сверточные структуры и более крупные скрытые слои. Это позволит проверить, сохраняется ли наблюдаемое преимущество модифицированного Latency-энкодера по вычислительной эффективности при усложнении задачи.

В-четвертых, важным направлением является исследование более сложных способов входного кодирования, которые в настоящей работе только обсуждались концептуально, но не были реализованы в экспериментальном протоколе. К ним относятся кодирование по изменениям, сканирующие режимы подачи изображения и гибридные схемы, сочетающие ранговую и частотную информацию.

В-пятых, представляет интерес дальнейшее развитие этапа обратного отображения активности сети в метки классов, включая сравнение различных способов построения карты меток нейронов, более сложные схемы голосования по активности скрытого слоя и использование признаков на основе счетчиков импульсов в сочетании с внешними классификаторами.

Таким образом, будущая работа должна объединить три направления: более широкий класс энкодеров, более глубокий анализ параметров и полную оценку итогового качества распознавания, что позволит перейти от анализа динамики обучения к построению практически применимого протокола выбора кодировщика для спайковых сетей.

ЗАКЛЮЧЕНИЕ

Исследовано влияние способа кодирования статического изображения в импульсную последовательность на динамику обучения спайковой нейронной сети. Рассмотрены два основных семейства кодирования, реализованные в экспериментальном протоколе: кодирование по времени первого импульса (Latency) и пуассоновское кодирование по интенсивности (Poisson) в стохастическом и детерминированном режимах. Показано, что при фиксированной архитектуре сети и одинаковых базовых параметрах обучения выбор кодировщика существенно влияет не только на плотность активности скрытого слоя, но и на вычислительную стоимость обработки, выраженную через число синаптических операций и интегральный энергетический прокси.

Ключевым результатом работы стало введение модифицированного латентностного кодировщика с порогом тишины $x_{\min} = 0$, позволяющего подавлять фоновые пиксели и устранять нефункциональный «залп» событий на последнем шаге времени. Экспериментально показано, что такая модификация практиче-

ски не ухудшает полезную выходную активность сети, но резко снижает вычислительную стоимость по сравнению с базовым вариантом Latency без фильтрации фона. В сравнении с пуассоновскими режимами модифицированный Latency продемонстрировал наиболее выгодное соотношение между информативной активностью и стоимостью вычислений, тогда как стохастический и детерминированный Poisson обеспечили режим работы, более плотный, но и более затратный по событиям. При этом детерминированный пуассоновский режим показал практически те же свойства, что и стохастический, оставаясь удобным воспроизводимым вариантом для систематических сравнений.

Таким образом, полученные результаты подтверждают, что выбор энкодера является не второстепенной технической деталью, а одним из центральных факторов, определяющих режим обучения спайковой сети. Предложенный подход позволяет рассматривать кодирование входа как самостоятельный объект оптимизации и сравнивать различные режимы по метрикам «активность – стоимость – специализация нейронов». Это создает основу для дальнейшего перехода от анализа динамики обучения к построению полного протокола распознавания, включающего карту меток нейронов, выходной классификатор и итоговую оценку точности.

Доступность данных и кода

Использован открытый набор данных MNIST [7]. Код экспериментов реализован в виде вычислительного colab-ноутбука и доступен по ссылке https://github.com/alexander-toshev/science-reports/tree/main/energy_efficient.

Благодарности

Работа выполнена за счет гранта, предоставленного Академией наук Республики Татарстан образовательным организациям высшего образования, научным и иным организациям на поддержку планов развития кадрового потенциала в части стимулирования их научных и научно-педагогических работников к защите докторских диссертаций и выполнению научно-исследовательских работ (Соглашение от 22.12.2025 № 12/2025-ПД-КФУ)

Конфликт интересов

Автор заявляет об отсутствии конфликта интересов.

СПИСОК ЛИТЕРАТУРЫ

1. *Gerstner W., Kistler W.* Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge: Cambridge University Press, 2002.
<https://doi.org/10.1017/CBO9780511815706>
2. *Maass W.* Networks of Spiking Neurons: The Third Generation of Neural Network Models // *Neural Networks*. 1997. Vol. 10, No. 9. P. 1659–1671.
[https://doi.org/10.1016/S0893-6080\(97\)00011-7](https://doi.org/10.1016/S0893-6080(97)00011-7)
3. *Davies M. et al.* Loihi: A Neuromorphic Manycore Processor with On-Chip Learning // *IEEE Micro*. 2018. Vol. 38, No. 1. P. 82–99.
<https://doi.org/10.1109/MM.2018.112130359>
4. *Diehl P.U., Cook M.* Unsupervised Learning of Digit Recognition Using Spike-Timing-Dependent Plasticity // *Frontiers in Computational Neuroscience*. 2015. Vol. 9. Art. 99. <https://doi.org/10.3389/fncom.2015.00099>
5. *Rueckauer B. et al.* Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification // *Frontiers in Neuroscience*. 2017. Vol. 11. Art. 682.
<https://doi.org/10.3389/fnins.2017.00682>
6. *Guo W. et al.* Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems // *Frontiers in Neuroscience*. 2021. Vol. 15. Art. 638474. <https://doi.org/10.3389/fnins.2021.638474>
7. *LeCun Y., Bottou L., Bengio Y., Haffner P.* Gradient-Based Learning Applied to Document Recognition // *Proceedings of the IEEE*. 1998. Vol. 86, No. 11. P. 2278–2324. <https://doi.org/10.1109/5.726791>
8. *Izhikevich E.M.* Which Model to Use for Cortical Spiking Neurons? // *IEEE Transactions on Neural Networks*. 2004. Vol. 15, No. 5. P. 1063–1070.
<https://doi.org/10.1109/TNN.2004.832719>
9. *Caporale N., Dan Y.* Spike Timing-Dependent Plasticity: A Hebbian Learning Rule // *Annual Review of Neuroscience*. 2008. Vol. 31. P. 25–46.
<https://doi.org/10.1146/annurev.neuro.31.060407.125639>
10. *Thorpe S., Delorme A., Van Rullen R.* Spike-Based Strategies for Rapid Processing // *Neural Networks*. 2001. Vol. 14, No. 6–7. P. 715–725.
[https://doi.org/10.1016/S0893-6080\(01\)00083-1](https://doi.org/10.1016/S0893-6080(01)00083-1)

11. Hazan H. et al. BindsNET: A Machine Learning-Oriented Spiking Neural Networks Library in Python // *Frontiers in Neuroinformatics*. 2018. Vol. 12. Art. 89. <https://doi.org/10.3389/fninf.2018.00089>

GENERATING TEMPORAL SIGNALS FROM STATIC IMAGES FOR SPIKING NEURAL NETWORKS

A. S. Toshchev^[0000-0003-4424-6822]

Kazan Federal University, Kazan, Russia

atoschev@kpfu.ru

Abstract

Spiking neural networks (SNNs), i.e., neural architectures that represent and transmit information in the form of temporally distributed spikes, require time-dependent input, whereas data in computer vision are most commonly available as static images. This study addresses the transformation pipeline “image → temporal signal → spikes” and examines how the choice of input encoding influences SNN training dynamics, spike activity density, and computational cost. The experimental section implements and compares two encoding families: first-spike-time encoding (Latency) and intensity-based Poisson encoding (Poisson). Within these families, four operating modes are considered: baseline Latency without background suppression, modified Latency with a silence threshold, stochastic Poisson, and deterministic Poisson. The evaluation employs the following metrics: the average number of spikes per sample, the number of synaptic operations, an energy-related proxy metric, and indicators characterizing competition among hidden-layer neurons. Experiments conducted on the MNIST dataset (60000 training and 10000 test images) using a network with a hidden layer of 100 neurons and a simulation horizon of 200 time steps demonstrate that all examined modes support stable learning without activity collapse. Among them, the modified Latency mode with a silence threshold of $x_{\min} = 0.05$ achieves the most favorable balance between useful activity and computational cost: at 323.41 spikes per sample, it requires 14925.09 synaptic operations, whereas the baseline Latency mode without background filtering, despite exhibiting a comparable

level of output activity (311.22 spikes per sample), requires 78400 synaptic operations.

Keywords: *spiking neural networks, image recognition, signal encoding, image encoding.*

REFERENCES

1. Gerstner W., Kistler W. Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge: Cambridge University Press, 2002.

<https://doi.org/10.1017/CBO9780511815706>

2. Maass W. Networks of Spiking Neurons: The Third Generation of Neural Network Models // Neural Networks. 1997. Vol. 10, No. 9. P. 1659–1671.

[https://doi.org/10.1016/S0893-6080\(97\)00011-7](https://doi.org/10.1016/S0893-6080(97)00011-7)

3. Davies M. et al. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning // IEEE Micro. 2018. Vol. 38, No. 1. P. 82–99.

<https://doi.org/10.1109/MM.2018.112130359>

4. Diehl P.U., Cook M. Unsupervised Learning of Digit Recognition Using Spike-Timing-Dependent Plasticity // Frontiers in Computational Neuroscience. 2015. Vol. 9. Art. 99. <https://doi.org/10.3389/fncom.2015.00099>

5. Rueckauer B. et al. Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification // Frontiers in Neuroscience. 2017. Vol. 11. Art. 682. <https://doi.org/10.3389/fnins.2017.00682>

6. Guo W. et al. Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems // Frontiers in Neuroscience. 2021. Vol. 15. Art. 638474. <https://doi.org/10.3389/fnins.2021.638474>

7. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-Based Learning Applied to Document Recognition // Proceedings of the IEEE. 1998. Vol. 86, No. 11. P. 2278–2324. <https://doi.org/10.1109/5.726791>

8. Izhikevich E.M. Which Model to Use for Cortical Spiking Neurons? // IEEE Transactions on Neural Networks. 2004. Vol. 15, No. 5. P. 1063–1070. <https://doi.org/10.1109/TNN.2004.832719>

9. Caporale N., Dan Y. Spike Timing-Dependent Plasticity: A Hebbian Learning Rule // Annual Review of Neuroscience. 2008. Vol. 31. P. 25–46. <https://doi.org/10.1146/annurev.neuro.31.060407.125639>

10. *Thorpe S., Delorme A., Van Rullen R.* Spike-Based Strategies for Rapid Processing // *Neural Networks*. 2001. Vol. 14, No. 6–7. P. 715–725.

[https://doi.org/10.1016/S0893-6080\(01\)00083-1](https://doi.org/10.1016/S0893-6080(01)00083-1)

11. *Hazan H. et al.* BindsNET: A Machine Learning-Oriented Spiking Neural Networks Library in Python // *Frontiers in Neuroinformatics*. 2018. Vol. 12. Art. 89.

<https://doi.org/10.3389/fninf.2018.00089>

СВЕДЕНИЯ ОБ АВТОРЕ



ТОЩЕВ Александр Сергеевич – заведующий кафедрой, канд. техн. наук, доцент, КФУ / Институт информационных технологий и интеллектуальных систем / Кафедра цифровой аналитики и технологий искусственного интеллекта, г. Казань.

Aleksandr Sergeevich TOSHCHEV – Department Head, Ph.D. in Engineering, associate professor at Kazan Federal University (KFU), Institute of Information Technologies and Intelligent Systems, Department of Digital Analytics and Artificial Intelligence Technologies, Kazan.

email: atoschev@kpfu.ru

ORCID: 0000-0003-4424-6822

Материал поступил в редакцию 16 марта 2026 года