

УДК: 004.032.26

Метод адаптивных гауссовых рецептивных полей для спайкового кодирования числовых переменных

М. В. Киселев^a, А. М. Урусов^b, А. Ю. Иваницкий^c

Лаборатория нейроморфных вычислений, Чувашский государственный университет им. И. Н. Ульянова,
Россия, 428018, г. Чебоксары, Московский проспект, д. 15

E-mail: ^a mkiselev@chuvsu.ru, ^b am.urusov@chuvsu.ru, ^c ivanitskiy@hotmail.com

Получено 26.02.2025, после доработки — 20.05.2025.

Принято к публикации 04.06.2025.

Одна из серьезных проблем, ограничивающих применение импульсных нейронных сетей в прикладных информационных системах, — это кодирование числовых данных в виде последовательностей спайков — бескачественных атомарных объектов, которыми обмениваются нейроны в импульсных нейросетях. Особенно остро эта проблема стоит в задачах обучения с подкреплением агентов, функционирующих в динамичном реальном мире, так как кроме точности кодирования надо учитывать еще его динамические характеристики. Одним из распространенных является метод кодирования гауссовыми рецептивными полями (ГРП). В этом методе одна числовая переменная, подаваемая на вход импульсной нейронной сети, представляется потоками спайков, испускаемых некоторым количеством входных узлов сети. При этом частота генерации спайков каждым входным узлом отражает близость текущего значения этой переменной к значению — центру рецептивного поля, соответствующего данному входному узлу. В стандартном методе ГРП центры рецептивных полей расположены эквидистантно. Это оказывается неэффективным в случае очень неравномерного распределения кодируемой величины. В настоящей работе предлагается усовершенствование этого метода, основанное на адаптивном выборе центров рецептивных полей и вычислении частот потоков спайков. Производится сравнение предлагаемого усовершенствованного метода ГРП с его стандартным вариантом с точки зрения объема сохраняемой при кодировании информации и с точки зрения точности классификационной модели, построенной на закодированных в виде спайков данных. Доля сохраняемой при спайковом кодировании информации для стандартного и адаптивного ГРП оценивается с помощью процедуры прямого и обратного кодирования большой выборки числовых значений из треугольного распределения вероятности и сравнения числа совпадающих бит в исходной и восстановленной выборке. Сравнение на основе точности классификации проводилось на задаче оценки текущего состояния, возникающей при реализации обучения с подкреплением. При этом классификационные модели строились тремя принципиально различными алгоритмами машинного обучения — алгоритмом ближайших соседей, случайным лесом решений и многослойным перцептроном. В статье демонстрируется преимущество предложенного нами метода во всех проведенных тестах.

Ключевые слова: импульсные нейронные сети, гауссовы рецептивные поля, спайковое кодирование информации

Работа выполнена при финансовой поддержке Российского научного фонда (грант 25-21-00126).

UDC: 004.032.26

The adaptive Gaussian receptive fields for spiking encoding of numeric variables

M. V. Kiselev^a, A. M. Urusov^b, A. Yu. Ivanitsky^c

Laboratory of neuromorphic computations, Chuvash State University,
15 Moskovsky prospect, Cheboksary, 428018, Russia

E-mail: ^a mkiselev@chuvsu.ru, ^b am.urusov@chuvsu.ru, ^c ivanitskiy@hotmail.com

Received 26.02.2025, after completion — 20.05.2025.

Accepted for publication 04.06.2025.

Conversion of numeric data to the spiking form and information losses in this process are serious problems limiting usage of spiking neural networks in applied informational systems. While physical values are represented by numbers, internal representation of information inside spiking neural networks is based on spikes — elementary objects emitted and processed by neurons. This problem is especially hard in the reinforcement learning applications where an agent should learn to behave in the dynamic real world because beside the accuracy of the encoding method, its dynamic characteristics should be considered as well. The encoding algorithm based on the Gaussian receptive fields (GRF) is frequently used. In this method, one numeric variable fed to the network is represented by spike streams emitted by a certain set of network input nodes. The spike frequency in each stream is determined by proximity of the current variable value to the center of the receptive field corresponding to the given input node. In the standard GRF algorithm, the receptive field centers are placed equidistantly. However, it is inefficient in the case of very uneven distribution of the variable encoded. In the present paper, an improved version of this method is proposed which is based on adaptive selection of the Gaussian centers and spike stream frequencies. This improved GRF algorithm is compared with its standard version in terms of amount of information lost in the coding process and of accuracy of classification models built on spike-encoded data. The fraction of information retained in the process of the standard and adaptive GRF encoding is estimated using the direct and reverse encoding procedures applied to a large sample from the triangular probability distribution and counting coinciding bits in the original and restored samples. The comparison based on classification was performed on a task of evaluation of current state in reinforcement learning. For this purpose, the classification models were created by machine learning algorithms of very different nature — nearest neighbors algorithm, random forest and multi-layer perceptron. Superiority of our approach is demonstrated on all these tests.

Keywords: spiking neural networks, Gaussian receptive fields, spiking information encoding

Citation: *Computer Research and Modeling*, 2025, vol. 17, no. 3, pp. 389–400 (Russian).

The work was supported by Russian Scientific Foundation (Grant 25-21-00126).

1. Введение

В процессе развития технологий искусственного интеллекта, основанного на нейронных сетях глубинного обучения, трансформерах, диффузионных моделях и т. д., их применение во всех сферах жизни делает все заметнее внутренне присущие этим технологиям проблемы и ограничения. Прежде всего они связаны с большим энергопотреблением вычислительных систем, реализующих обучение и применение искусственных нейронных сетей (ИНС) этого типа. Это, с одной стороны, делает обучение больших ИНС в масштабе всего мира одной из существенных статей расхода энергоресурсов человечества, а с другой — препятствует широкому использованию таких сетей в интеллектуальных автономных устройствах (например, в робототехнике), лишенных постоянного доступа к мощному источнику энергии. По мнению многих исследователей, путем преодоления этой проблемы может быть применение так называемых нейроморфных технологий, основанных на импульсных нейронных сетях (ИмНС) и специализированных нейропроцессорах. Одним из принципиальных отличий ИмНС от ИНС является использование представления обрабатываемой информации не в виде чисел, а в виде потоков элементарных бескачественных событий, так называемых спайков. Возникающая таким образом возможность построить функционирование сети как вычисления, основанные на событиях (event-based computations), причем не включающие энергозатратные операции умножения и вычисления трансцендентных функций и асинхронный массивно-параллельный характер этих вычислений дают возможность сократить энергопотребление на много порядков [Intel benchmarks. . . , 2020; Ivanov et al., 2022].

Однако спайковое представление информации, будучи ключом к энергоэкономичности интеллектуальных систем, вызывает, с другой стороны, ряд специфических проблем, главная из которых — проблема конвертации числовых данных в спайковую форму. Существующие информационные системы работают с числовой информацией, поэтому для использования их в рамках ИмНС требуется реализовать какой-либо метод кодирования (и декодирования) числовых величин в спайки.

Таких методов было предложено несколько. Их обзор и сравнительный анализ содержатся, например, в работах [Auge et al., 2021; Guo et al., 2021; Yamazaki et al., 2022]. Хотя методы спайкового кодирования достаточно разнообразны, они сводятся к трем основным принципам.

- 1. Частотное кодирование.** В этом методе значение числовой величины кодируется числом спайков в единицу времени. В простейшем варианте кодирование осуществляется одним нейроном. Это очень неточный и медленный метод кодирования. Действительно, чтобы закодировать числовую величину с достаточной точностью при ограниченной частоте спайков требуется значительный временной интервал для измерения средней частоты спайков, поэтому кодирование быстро меняющихся величин становится невозможным. Эта проблема отчасти решается популяционно-частотным кодированием, где считается число спайков не одного нейрона, а по значительной популяции. Однако здесь возникает проблема экономичности: отводить большой группе нейронов роль кодировщика одной числовой переменной нерационально. Особенно явной эта проблема делается в контексте обучения: если какая-то величина кодируется 30 нейронами, то в силу того, что с точки зрения кодирования все эти нейроны одинаковы, в процессе обучения веса 30 их постсинаптических связей с каким-либо другим нейроном должны изменяться одинаковым образом. Кроме того, этот метод неэкономичен и в терминах числа спайков — для точного кодирования какой-либо величины может потребоваться много спайков. Упомянем и еще один значительный недостаток — неточность декодирования. Трудно построить нейросетевую структуру, которая избирательно реагировала бы на входной поток спайков при условии средней частоты

спайков, лежащей в узком диапазоне. Все эти проблемы приводят к тому, что такой метод обычно применяется в сочетании с другими принципами кодирования.

2. **Кодирование латентностью, или временное кодирование.** В отличие от предыдущего подхода, в этой схеме кодирования важно точное положение спайков на временной оси. Обычно это положение отсчитывается от некоторых референтных моментов времени. Например, если кодируется вектор величин, или изображение, или еще какой-то статический объект, то на предъявления каждого такого объекта отводится некоторое время. Для кодирования некоторой величины, характеризующей предъявленный объект (например, яркости пикселя изображения), отводится один нейрон. При этом значение этой величины кодируется временем срабатывания этого нейрона по отношению к началу презентации объекта — большему значению величины соответствует меньшая задержка. Главное достоинство этого метода — экономичность. Одной величине соответствуют один нейрон и один спайк. Декодирование также достаточно эффективно, учитывая, что импульсный нейрон работает как детектор совпадающих по времени спайков. Слабой стороной является то, что логика этого способа кодирования плохо подходит для кодирования динамических данных (видео вместо разрозненных картинок), так как там нет естественного референтного процесса, по отношению к которому можно было бы отсчитывать задержки. Кроме того, большая длина интервала времени, используемого для кодирования одного значения, делает кодирование быстро меняющихся величин невозможным.
3. **Позиционное кодирование.** В этом способе область значений переменной разбивается на интервалы. Каждому интервалу соответствует нейрон. Если нейрон испускает спайк, это означает, что значение переменной находится в соответствующем ему интервале. Такой метод кодирования имеет много преимуществ. Он подходит для кодирования быстро меняющихся динамических величин, экономичен, так как требует лишь одного спайка для однократного кодирования. Такое кодирование легко декодируется и интерпретируется. К недостаткам можно отнести малую точность кодирования при небольшом числе кодирующих нейронов. Кроме того, этот метод кодирования никак не использует тот факт, что кодируемая величина — число. Так как разным интервалам соответствуют разные нейроны, нет возможности определить, далеко или близко по значению находятся кодируемые ими интервалы.

В связи с наличием недостатков у всех рассмотренных методов на практике обычно применяется то или иное их сочетание. Чаще всего используется сочетание позиционного и временного кодирования или позиционного и частотного. Основывается это сочетание на так называемом алгоритме гауссовых рецептивных полей (ГРП, англ. GRF — Gaussian receptive fields). В этом методе в области значений кодируемой переменной x выделяется N точек c_i — *центров рецептивных полей*. Каждому рецептивному полю приписывается его *ширина* s_i . При этом вместо точного кодирования текущего значения величины x как таковой производится грубое кодирование N величин

$$v_i = A e^{-(x-c_i)^2/s_i^2}, \quad (1)$$

где A — константа. Кодирование осуществляется N нейронами, каждый из которых соответствует одному рецептивному полю: таким образом реализуется принцип позиционного кодирования. А вот v_i переводятся в спайковый вид с помощью частотного или временного кодирования. При этом достигается нужный компромисс между точностью, экономичностью и динамическими характеристиками кодирования. Для эффективности кодирования принципиально важен оптимальный выбор пар $\langle c_i, s_i \rangle$. Именно этому вопросу и посвящено данное исследование. Отметим, что, так как эта работа проводилась в рамках проекта по реализации обучения с подкреплением

в ИмНС, для нас в первую очередь было важно кодирование динамических величин. Поскольку, как это было отмечено выше, кодирование латентностью плохо подходит для этой цели, нами использовалось частотное кодирование v_i . Однако результаты данного исследования в равной степени могут быть распространены и на вариант с временным кодированием, так как мы оптимизируем информативность самих величин v_i , безотносительно того, как они дальше будут кодироваться.

Метод кодирования данных с использованием гауссовых рецептивных полей является важным инструментом в различных подходах к обучению ИмНС и применялся во многих исследованиях. В [Sboev et al., 2018] описывается метод обучения ИмНС с использованием синаптической пластичности, зависящей от времени (STDP) для задачи классификации, в которой входные данные кодируются с помощью ГРП: метод протестирован на известных датасетах, таких как Wisconsin Breast Cancer, и показал свою эффективность по сравнению с SVM, случайными лесами (random forest) и нейросетями с Adam-оптимизатором. В [Rybka et al., 2024] авторы применяют ГРП для кодирования информации при сравнении способов уменьшения связности для ИмНС с мемристивной пластичностью. В работе [Firouzi et al., 2013] представлена модель ИмНС для алгоритма ALM, использующего метод Ink Drop Spread (IDS) для извлечения нечетких знаний. IDS, вдохновленный нейробиологией, позволяет эффективно извлекать знания из сенсорных данных, при этом входные данные кодируются с использованием ГРП, что обеспечивает эффективную обработку информации с применением STDP. Особое внимание использованию ГРП уделяется в работе [Schliebs, Kasabov, 2013], которая является обзором эволюции архитектуры развивающихся ИмНС (eSNN), в статье рассмотрены различные вопросы, включая обработку пространственно-временных данных и оптимизацию параметров для повышения точности классификации и предсказаний. Использование ГРП для кодирования входных данных играет ключевую роль в повышении эффективности работы моделей eSNN. В статье [Gardner, Grüning, 2021] авторы предлагают метод контролируемого обучения многослойных ИмНС, демонстрирующий эффективность в задачах классификации, применяя его к нескольким наборам данных, включая MNIST, с использованием ГРП для кодирования. В [Bohte et al., 2002a] авторами представлено правило обучения с учителем SpikeProp для ИмНС, показана эффективность при выполнении задачи сложной нелинейной классификации с временным кодированием, при использовании ГРП, те же авторы в [Bohte et al., 2002b] показывают эффективность ИмНС для решения задачи кластеризации без учителя при использовании ГРП для кодирования данных. В работе [Cui et al., 2025] авторы представляют биологически правдоподобную аналоговую вычислительную систему на фотоэлектронных синаптических транзисторах и мемристивной матрице для обработки движений человека, в которой входные данные кодируются ГРП в виде импульсных последовательностей. Наконец, в [Stromatias, 2011] описан процесс кодирования с применением ГРП в рамках популяционного кодирования, в этом подходе группа входных нейронов кодирует аналоговые данные. Во всех этих работах акцент делается на использовании ГРП для эффективного кодирования данных, что способствует улучшению обработки информации и оптимизации работы ИмНС.

В данной работе исследуется усовершенствованный метод ГРП. Предлагается сделать выбор центров и ширин кодирующих гауссиан адаптивным, зависящим от кодируемых данных, вместо системы одинаковых гауссиан с эквидистантными центрами, использованной в вышеупомянутых работах. Общая идея состоит в том, что центры гауссиан рассчитываются поквантильно, с учетом числа наблюдений, что обеспечивает более равномерное распределение кодируемой информации и улучшает точность кодирования особенно в случае неравномерного распределения значений кодируемых величин. Кроме того, для нахождения расстояний до центров рецептивных полей также используется величина, зависящая от распределения значений, вместо простой разности. Формально этот метод будет описан в следующем разделе. Его сравнение со

стандартным вариантом ГРП приведено в §§ 3 и 4. После них следуют обсуждение результатов и выводы.

2. Адаптивный алгоритм построения системы гауссовых рецептивных полей

Итак, нам требуется закодировать в виде потоков спайков некоторую числовую переменную x . В нашем методе не важно, является ли она непрерывно изменяющимся временным рядом, результатом измерений некоторого динамического процесса или полем базы данных, где записи могут относиться к совсем разным объектам. В последнем случае предполагается, что каждая запись (точнее, ее спайковое представление) предъявляется нейросети в течение некоторого времени, достаточного для получения определенного количества спайков. Единственно, что предполагается в нашем подходе, это наличие некоторой достаточно репрезентативной выборки значений этой переменной, дающей возможность судить об истинном распределении x . Обозначим эту выборку как $\{x_a\}$, $1 \leq a \leq n$.

Множество $\{x_a\}$ применяется для нахождения N центров рецептивных полей c_i . Выбор самого N в первую очередь влияет на точность кодирования и, как всегда в подобных случаях, делается на основе компромисса между точностью и имеющимися ресурсами (в данном случае для моделирования ИмНС соответствующего размера). При заданном N значения c_i ($i \in [1, N]$) выбираются следующим образом: множество $\{x_a\}$ сортируется по возрастанию. После этого

$$c_i = \frac{x_{\lfloor (i-1)d \rfloor + 1} + x_{\lfloor id \rfloor}}{2}, \quad (2)$$

где $d = \frac{n}{N}$, т. е. c_i выбираются по квантильному принципу.

В случае когда переменная имеет несколько доминирующих значений, представленных в значительной части множества $\{x_a\}$, может случиться так, что какие-то соседние центры рецептивных полей будут совпадать. В этом случае при вычислении (2) во множестве $\{x_a\}$ оставляются только уникальные значения.

Для кодирования величин x понадобится задать еще два параметра: это максимальная частота спайков A и *относительная ширина рецептивного поля* S . Смысл первого достаточно очевиден: он задает частоту спайков, выпускаемых i -м входным узлом сети, если $x = c_i$. Так как моделирование ИмНС происходит обычно в дискретном времени, разумным выбором по умолчанию является $A = 1$, т. е. генерация спайка на каждом шаге эмуляции. Мы будем использовать именно это значение. Второй параметр определяет избирательность рецептивного поля. Выбор этого параметра мы обсудим в конце данного раздела.

Теперь опишем процесс кодирования. В этом процессе участвуют N переменных *потенциала* P_i — по одной для каждого из N входных узлов, кодирующих значение x . В начале все $P_i = 0$. В каждый шаг эмуляции для каждого i определяется *приращение потенциала* p_i по формуле, приведенной ниже. В каждый шаг значение P_i увеличивается на p_i . Если P_i становится равным или большим 1, P_i уменьшается на 1, и i -й входной узел выпускает спайк. Приращения потенциала определяются по формуле

$$p_i(x) = e^{-d_i^2(x)/S^2}, \quad (3)$$

аналогичной (1), но расстояние $d_i(x)$ от текущего значения x до центра i -го рецептивного поля также вычисляется адаптивно, учитывая наблюдаемое распределение x , а не просто как разность

значений:

$$d_i(x) = \begin{cases} 0, & \text{если } x \leq c_1 \ \& \ i = 1 \ \vee \ x \geq c_N \ \& \ i = N, \\ \frac{x - c_{i-1}}{c_i - c_{i-1}}, & \text{если } c_{i-1} \leq x \leq c_i, \\ \frac{x - c_i}{c_{i+1} - c_i} & \text{если } c_i \leq x \leq c_{i+1}, \\ d_{i-1}(x) + 1, & \text{если } x < c_{i-1}, \\ d_{i+1}(x) + 1, & \text{если } x > c_{i+1}, \end{cases} \quad (4)$$

т. е. мы считаем все расстояния между центрами соседних рецептивных полей единичными. Например, если $c_1 = 0$, $c_2 = 1$, $c_3 = 3$, то $d_1(-3) = 0$, $d_2(-3) = 1$, $d_3(-3) = 2$; $d_1(1,5) = 1,25$, $d_2(1,5) = 0,25$, $d_3(1,5) = 0,75$.

Нетрудно видеть, что при равномерном распределении x (3), (4) приближается к (1). Кроме того, мы принимаем расстояния от значений x , выходящих за пределы диапазона $[c_1, c_N]$, до центров соответствующих крайних рецептивных полей нулевыми. Это логично, так как иначе значения x , далекие от краев этого диапазона, вообще не приводили бы к генерации спайков. В этом случае логически более оправданна максимально интенсивная генерация спайков входным узлом, соответствующим ближайшему крайнему рецептивному полю.

Отметим, что так как «расстояния» d_i являются безразмерными, то и ширина рецептивных полей S безразмерна. Сильно меньшим 1 его делать нельзя — в этом случае величины, лежащие посередине между центрами соседних рецептивных полей, вообще не будут приводить к генерации спайков. Кроме того, часто бывает полезным сделать среднюю частоту спайков от всех входных узлов сети примерно постоянной, что требует значений S , близких к 1 или более. Большие значения S приведут к снижению точности кодирования: разница в активности входного узла, соответствующего рецептивному полю, ближайшему к x , и более далеких от него, будет столь невелика, что не позволит судить об истинном значении x . Разумным выбором, соответствующим компромиссу между постоянством частоты спайков и селективностью, является $S = 0,6$. При таком выборе при $N = 10$ средняя суммарная частота спайков от всех рецептивных полей при любых значениях x варьирует не более чем на 13%. При этом доля спайков рецептивного поля с центром, совпадающим с текущим значением x , составляет не менее 93% от спайков всех рецептивных полей. Именно это значение мы будем использовать в нашей работе.

В следующих разделах мы рассмотрим преимущества предлагаемого способа кодирования на нескольких примерах.

3. Оценка объема информации, сохраняемой при спайковом кодировании, в случае треугольного распределения кодируемой величины

Основное требование к любой процедуре, преобразующей данные из одной формы в другую, является сохранение максимального объема информации об исходных данных. Оценка этой информации не всегда очевидна, особенно в случае, как это имеет место в данном исследовании, когда эти формы сильно различны. Одним из способов такой оценки является применение обратного преобразования и сравнение его результата с исходным массивом данных в терминах доли правильно воспроизведенной информации. Такой способ вряд ли позволит оценить долю информации, сохраняемой в процессе прямого преобразования, но как способ сравнения двух разных алгоритмов вполне разумен (при условии, что обратный алгоритм в обоих случаях один и тот же). Именно эту методологию мы и применили для сравнения стандартного и нашего варианта ГРП.

Очевидно, что два эти варианта ГРП совпадают при равномерном распределении кодируемой величины. Однако это распределение редко встречается в практически интересных задачах. Гораздо более обычными являются сильно отличающиеся от равномерного распределения — нормальное или еще более отличное от него логнормальное. Для сравнительного анализа мы выбрали умеренно неравномерное распределение — треугольное. Треугольное распределение определено на отрезке $[0, a]$ с плотностью вероятности $p(x) = \frac{a - |\frac{a}{2} - x|}{a^2}$.

Для численного анализа мы будем применять большие выборки из этого распределения — 100 000 значений. Таким образом, в (2) $n = 100\,000$. Центры рецептивных полей для стандартного ГРП выбираются эквидистантно — $c_i = \frac{(2i-1)a}{2N}$. Так как расстояние между центрами соседних рецептивных полей в стандартном ГРП равно $\frac{a}{N}$, то ширина рецептивных полей будет равна $s_i = \frac{5a}{N}$. Пусть одно значение кодируемой величины представляется $N_s = 10$ шагами эмуляции ИмНС.

В обоих случаях будет выбрана следующая естественная декодирующая процедура. Каждый спайк в спайковом представлении числовой величины x «голосует» за центр своего рецептивного поля. Таким образом, если входной узел, соответствующий i -му рецептивному полю, за 10 шагов испустил n_i спайков, восстановленное значение x будет равно

$$\bar{x} = \frac{\sum_i n_i c_i}{\sum_i n_i}.$$

Оценка объема восстановленной информации производится следующим образом. Пусть мы кодируем достаточно высокоразрядные целые числа, например по 10 бит на число, т. е. $a = 1024$. Пусть кодировалось число x , а декодировано было \bar{x} . Тогда для числа x правильно было декодировано столько бит, сколько совпадает старших бит в бинарном представлении x и \bar{x} , начиная со старшей битовой позиции в числах x и \bar{x} . Например, если $x = 108$ (1101100), а $\bar{x} = 103$ (1100111), то правильно восстановленных бит — 3. При некоторой произвольности данного метода, очевидно, что для больших выборок такой подсчет восстановленной информации будет вполне показателен, особенно когда он используется не для вычисления самого этого значения, а для сравнения двух методов кодирования.

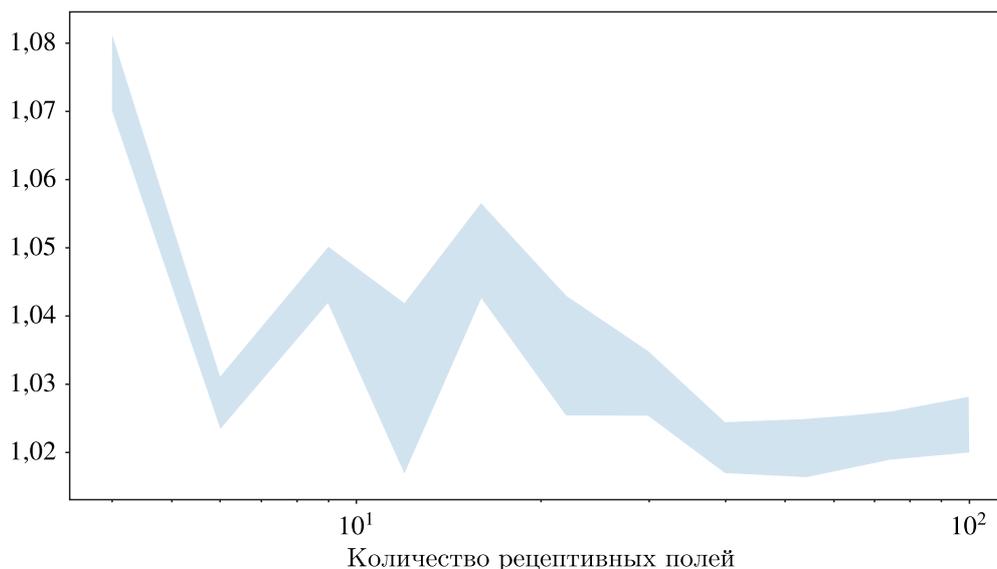


Рис. 1. Отношение объема информации о первоначальной числовой выборке для адаптивного и стандартного методов ГРП при разном числе рецептивных полей. Показана зона шириной в два стандартных отклонения для 10 численных экспериментов

Было измерено отношение объема восстановленной информации для выборок из 100 000 целых чисел, подчиняющихся треугольному распределению с $a = 1024$, для разных значений N в диапазоне [5, 100]. Для каждого N ставилось 10 численных экспериментов.

Результаты представлены на рис. 1. Видно, что для всех N адаптивный метод ГРП сохраняет больше информации, чем стандартный. Эта разница, хотя и невелика (около 4% в среднем), статистически значима. Данный результат легко верифицируем — в [Github] приведена соответствующая программа на Python. Подчеркнем, что разница между рассматриваемыми алгоритмами может быть кардинально больше для часто встречающихся сильно неравномерных распределений, например логнормального.

В следующем разделе мы сравним стандартный и адаптивный методы ГРП в терминах точности классификационных моделей, построенных на спайковом представлении данных.

4. Сравнение алгоритмов кодирования на задаче определения целевого состояния в обучении с подкреплением

Так как данное исследование выполнялось в контексте проекта по реализации обучения с подкреплением в ИмНС, важно было протестировать предложенный метод спайкового кодирования именно в этой области. В качестве тестовой задачи была выбрана простая эмуляция игры в пинг-понг, входящая в набор тестов по обучению с подкреплением ATARI [Atari]. В этой задаче нейросеть, управляющая ракеткой размера R , должна учиться отбивать мячик, летающий по квадратному полю единичного размера; сама ракетка может двигаться вертикально по левой границе этого поля (рис. 2). Один из элементов этой задачи — определение целевого состояния, т. е. такого положения ракетки, при котором ей уже не нужно двигаться, чтобы отбить шарик. Нейросеть получает непрерывную информацию о пяти величинах: двумерных координатах шарика $-0,5 \leq x, y \leq 0,5$, двух компонентах его скорости v_x, v_y и вертикальной координате центра ракетки y_r . Все эти величины кодируются потоками спайков. Для этого может быть применено как стандартное, так и адаптивное кодирование с помощью ГРП. Чтобы сравнить качество этих алгоритмов, мы решали задачу классификации целевого состояния на основе вышеупомянутых пяти величин.

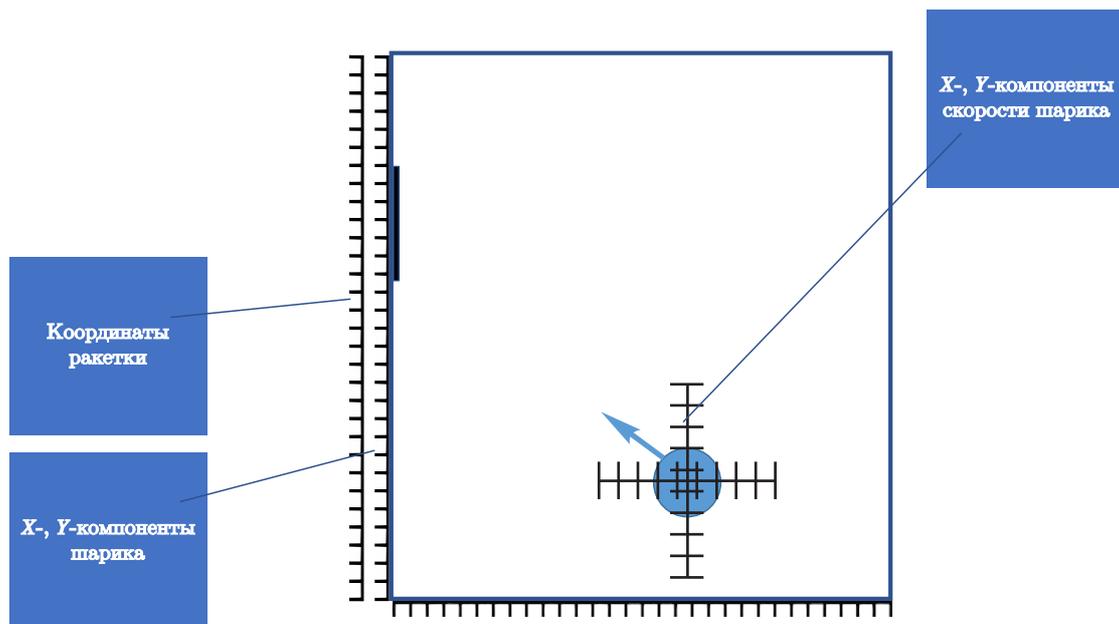


Рис. 2. Схематичное изображение мира игры в пинг-понг и сигналов, передаваемых на вход сети, — координат и компонент скорости шарика и вертикальной координаты ракетки

Правильный ответ может быть легко выражен аналитически: ракетка отобьет мячик, если

$$y_r + 0,5 - \frac{R}{2} < z < y_r + 0,5 + \frac{R}{2} \quad \vee \quad 1,5 - y_r - \frac{R}{2} < z < 1,5 - y_r + \frac{R}{2}, \quad (5)$$

где $z = w - 2 \left[\frac{w}{2} \right]$, где $w = y - \frac{(x+0,5)v_y}{v_x} + 0,5$. Здесь мы учитываем только случаи, когда шарик находится в левой половине квадрата и летит в сторону ракетки: $x < 0$ и $v_x < 0$. В этой задаче мы используем 10 рецептивных полей для каждой из 5 величин, но так как $x < 0$, $v_x < 0$, то для x и v_x используются только 5.

Для теста были взяты 621 652 состояний (всего эмуляция игры включала 2 000 000 состояний, но были отобраны только удовлетворяющие критерию из предыдущего абзаца). По формуле (5) были отмечены целевые состояния. Были сформированы три датасета: исходный, содержащий 5 действительных чисел в каждой записи (ORI); датасет, где эти параметры были закодированы с помощью традиционного ГПП (OLD); датасет, где эти параметры были закодированы с помощью адаптивного ГПП (NEW). Максимальное число спайков на одно значение на одну запись было равно 10. Таким образом, датасеты OLD и NEW включали $5 + 10 + 5 + 10 + 10 = 40$ числовых предикторов — целых чисел от 0 до 10.

Для сравнения точности моделей мы выбрали три максимально непохожих друг на друга алгоритма, основанных на совершенно разных математических принципах: алгоритм ближайших соседей, делающий классификацию на основе близости классифицируемой записи к записям из обучающего массива (KNN), случайный лес решений — ансамбль деревьев решений (RF) и многослойный перцептрон (MLP). В соответствии со стандартами машинного обучения была применена 5-кратная кросс-валидация — с одним и тем же разбиением для всех датасетов. Тем самым были получены ошибки классификации и стандартные отклонения для них. Результаты представлены в таблице 1.

Таблица 1. Ошибки классификационных моделей, полученных тремя разными алгоритмами машинного обучения: на исходном датасете, при использовании обычной и адаптивной кодировки ГПП

Датасет (метод кодирования)	KNN		RF		MLP	
	Средняя ошибка, %	Стандартное отклонение ошибки, %	Средняя ошибка, %	Стандартное отклонение ошибки, %	Средняя ошибка, %	Стандартное отклонение ошибки, %
ORI	9,07	0,04	0,26	0,01	8,94	0,39
OLD	1,10	0,02	1,03	0,03	2,23	0,02
NEW	1,04	0,03	0,97	0,01	2,11	0,06

Наблюдается увеличение точности моделей KNN и MLP на спайковых данных по сравнению с исходным датасетом, что кажется парадоксальным. Этот эффект объясняется более многомерным представлением данных в спайковых датасетах. Похожий эффект используется при применении метода машины опорных векторов (support vector machine) — переход к многомерному представлению делает линейно сепарабельными целевые классы, которые не были таковыми в низкоразмерном представлении. Но это, конечно, говорит и о том, что спайковая кодировка хорошо сохраняет информацию. Метод RF здесь имеет столь сильное преимущество, так как задача допускает точное решение, а объем обучающих данных велик, это позволяет ансамблю деревьев (информационно очень емкой модели) хранить много информации об обучающей выборке без опасности переобучения.

Если же обратиться к непосредственной теме данной статьи, то видно, что для всех трех алгоритмов предлагаемый адаптивный вариант ГПП показывает значимо лучшие результаты, чем стандартный вариант. Разница составляет несколько стандартных отклонений.

Выводы и заключение

Эффективное преобразование данных с сохранением максимального объема информации является ключевой задачей при использовании различных методов кодирования. В данном исследовании проведен сравнительный анализ классического и адаптивного методов кодирования с помощью ГРП. На примере треугольного распределения кодируемой числовой величины показано, что адаптивное кодирование позволяет сохранить больше информации, чем классическое эквидистантное расположение центров рецептивных полей. Независимо от числа нейронов адаптивный метод демонстрирует статистически значимое преимущество, обеспечивая в среднем на 4% более высокую точность восстановления данных. Кроме того, ожидается, что при более неравномерных распределениях данных, например логнормальном, разница между методами может быть значительно выше. Эти результаты подтверждают, что адаптивное расположение центров рецептивных полей и модифицированная оценка «расстояния» от кодируемой точки до этих центров позволяют более точно представлять исходные числовые величины в спайковой форме. Также адаптивный метод ГРП был протестирован в контексте задачи обучения с подкреплением. В качестве теста использовалась симуляция игры в пинг-понг, где целевой задачей является обучение нейросети управлять ракеткой для отбивания мячика. Результаты экспериментов показали, что адаптивное кодирование позволило добиться более высокой точности предсказания целевого состояния ракетки. При использовании трех различных методов машинного обучения (KNN, RF, MLP) во всех случаях модели, обученные на данных, закодированных адаптивным методом ГРП, показали более высокие показатели точности по сравнению с классическим подходом.

Таким образом, результаты исследования подтверждают эффективность адаптивного метода ГРП. Он не только позволяет лучше сохранять информацию при кодировании, но и улучшает качество обучения моделей, использующих закодированные данные. Особенно перспективным его применение представляется в задачах, связанных с обработкой динамических данных и обучением с подкреплением. При этом вычислительная сложность возрастает незначительно: дополнительные затраты на вычисление «расстояний» по формуле (4) пренебрежимо малы по сравнению с затратами на вычисление гауссиан, необходимое и в стандартном, и в нашем методе.

В дальнейших исследованиях планируется изучить применение адаптивного метода ГРП в более сложных задачах обучения с подкреплением, а также в задачах, требующих обработки временных зависимостей.

Список литературы (References)

- Atari. — <https://en.wikipedia.org/wiki/Atari> (data access: 03.06.2025).
- Auge D., Hille J., Mueller E., Knoll A. A survey of encoding techniques for signal processing in spiking neural networks // *Neural Process. Lett.* — 2021. — Vol. 53. — P. 4693–4710.
- Bohte S. M., Kok J. N., Poutre H. Error-backpropagation in temporally encoded networks of spiking neurons // *Neurocomputing.* — 2002a. — Vol. 48. — P. 17–37.
- Bohte S. M., Poutre H., Kok J. N. Unsupervised clustering with spiking neurons by sparse temporal coding and multilayer RBF networks // *IEEE Trans. Neural Netw.* — 2002b. — Vol. 13, No. 2. — P. 426–435.
- Cui H., Xiao Y., Yang Y., Pei M., Ke Sh., Fang X., Qiao L., Shi K., Long H., Xu W., Cai P., Lin P., Shi Y., Wan Q., Wan Ch. A bioinspired in-materia analog photoelectronic reservoir computing for human action processing // *Nature Communications.* — 2025. — Vol. 16. — P. 2263. — <https://doi.org/10.1038/s41467-025-56899-3>

- Firouzi M., Shouraki S.B., Rostami M.G.* Spiking neural network ink drop spread, Spike-IDS // Yamaguchi Y. (ed.) *Advances in cognitive neurodynamics (III)*. — Dordrecht: Springer, 2013. — https://doi.org/10.1007/978-94-007-4792-0_9
- Gardner B., Grüning A.* Supervised learning with first-to-spike decoding in multilayer spiking neural networks // *Front. Comput. Neurosci.* — 2021. — Vol. 15. — P. 617862. — DOI: 10.3389/fncom.2021.617862
- GitHub. Adaptive Gaussian receptive fields for SNN: Source code and scripts. — <https://github.com/AndreyUrus/Adaptive-Gaussian-Receptive-Fields-SNN> (data access: 03.06.2025).
- Guo W., Fouda M.E., Eltawil A.M., Salama K.N.* Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems // *Front. Neurosci.* — 2021. — Vol. 15. — P. 638474. — DOI: 10.3389/fnins.2021.638474
- Intel benchmarks for Loihi neuromorphic computing chip. — 2020. — <https://www.eetasia.com/intel-benchmarks-for-loihi-neuromorphic-computing-chip/> (data access: 03.06.2025).
- Ivanov D., Chezhegov A., Kiselev M., Grunin A., Larionov D.* Neuromorphic artificial intelligence systems // *Frontiers in Neuroscience*. — 2022. — Vol. 16. — P. 1513. — DOI: 10.3389/fnins.2022.959626
- Rybka R., Davydov Yu., Vlasov D., Serenko A., Sboev A., Ilyin V.* Comparison of bagging and sparsity methods for connectivity reduction in spiking neural networks with memristive plasticity // *Big Data Cogn. Comput.* — 2024. — Vol. 8. — P. 22.
- Sboev A., Vlasov D., Rybka R., Serenko A.* Spiking neural network reinforcement learning method based on temporal coding and STDP // *Procedia Comput. Sci.* — 2018. — Vol. 145. — P. 458–463.
- Schliebs S., Kasabov N.* Evolving spiking neural network — a survey // *Evolving Systems*. — 2013. — Vol. 4. — P. 87–98.
- Stromatias E.* Developing a supervised training algorithm for limited precision feed-forward spiking neural networks // *ArXiv e-prints*. — 2011. — <http://adsabs.harvard.edu/abs/2011arXiv1109.2788S>
- Yamazaki K., Vo-Ho V.-K., Bulsara D., Le N.* Spiking neural networks and their applications: A review // *Brain Sciences*. — 2022. — Vol. 12, No. 7. — P. 863.