

**Импульсная нейронная сеть на основе модели «ключ-порог»**

**Малявко А.А.<sup>1</sup>, Гаврилов А.В.<sup>1</sup>**

Новосибирский государственный технический университет<sup>1</sup>

*В статье предлагается новая модель импульсной нейронной сети, основанной на пороговом распознавании ключа (key threshold spike neuron network – KTSNN). Эта сеть состоит из нейронов, ориентированных на распознавание значений входного вектора как ключа, хранящегося в нейроне. Предлагается программная реализация этой модели. Обсуждаются возможные методы обучения, реализации и использования этой модели.*

**Ключевые слова:** нейронная сеть, импульсный нейрон, алгоритмы обучения.

В качестве модели импульсного нейрона обычно рассматривается пороговое устройство, тем или иным способом интегрирующее взвешенные входные сигналы (как правило – с утечкой, т.е. уменьшением накапливаемого значения при отсутствии входных импульсов) и срабатывающее (формирующее один выходной импульс или серию импульсов) в тот момент, когда величина интеграла от значений входов достигает установленного порога [1]. Некоторое количество таких нейронов каким-либо образом соединяются друг с другом, образуя нейронную сеть. Сети, построенные из нейронов подобного типа, используются в частности для решения задач классификации объектов. При этом выявлением классифицирующих признаков занимаются ансамбли нейронов, формируемые при обучении сети.

Предлагается модель импульсной сети на основе модели «ключ-порог», предложенной в [2] для статических нейронов, в которой каждый нейрон может играть роль целого ансамбля пороговых нейронов, т.е. по существу – персептрона. Для этого активирующая функция вместо суммирования во времени значений входных сигналов вычисляет степень их близости заданному ключевому пороговому значению нейрона. Нейрон срабатывает только тогда, когда паттерн (распределение входных значений либо во времени, либо в пространстве) совпадает с его ключевым кодом или отличается от него не более, чем на величину порога.

Нейроны модели «ключ-порог» могут быть двух типов. S-нейроны ориентированы на распознавание значений ключевых входов, распределенных в пространстве (совокупность спайков на нескольких синапсах). Т-нейроны обрабатывают сигналы, распределенные во времени – последовательность спайков на единственном синапсе.

Сеть, состоит из трех слоев: входного, скрытого и выходного.

Нейроны входного слоя генерируют импульсы в соответствии со значением некоторого параметра внешней среды, воспринимаемого датчиком, например при обнаружении препятствия. Скрытый слой состоит из S- или Т-нейронов, которые отличаются от привычных моделей нейронов тем, что не имеют весов и взвешенного суммирования входных сигналов. Принципом работы такого нейрона является сопоставление - двоичного кода – так называемого «ключа»  $k=(k_1, \dots, k_n)$  с входным кодом  $x=(x_1, \dots, x_n)$  совокупности сигналов поступивших на его вход (входы). При совпадении этих кодов нейрон срабатывает и генерирует выходной импульс (спайк). Длина ключа  $n$  определяет количество входных нейронов, с которыми может быть связан по входам нейрон скрытого слоя. Если входных нейронов больше, чем входов нейрона, входы скрытого нейрона случайно распределяются по входным нейронам при инициализации сети.

Параметрами сети, изменяемыми в процессе обучения, являются ключи выходных и скрытых нейронов  $k_i=(0, 1)$ .

Ключ каждого нейрона выходного слоя меняется в соответствии со следующим правилом:

Обозначим:

$y$  – выход нейрона ( $y=1$  означает, что нейрон сработал);

$g$  – желаемое значение выхода нейрона.

**If** ( $y=1$  и  $g=0$ ) или ( $y=0$  и  $g=1$ )

**Then** случайно выбирается и инвертируется разряд ключа  $k_s$ .

Применение такого алгоритма обучения выходного нейрона приводит к уменьшению вероятности ложного срабатывания или несрабатывания нейрона в дальнейшем при таких же значениях выходов нейронов скрытого слоя.

Для обучения нейронов скрытого слоя может быть использован следующий алгоритм.

Случайно выбирается скрытый нейрон, связанный по выходу с  $j$ -м входом выходного нейрона такой, что  $j \neq s$ . Для него применяется выше описанное правило, но при этом в качестве  $g$  выступает значение  $k_j$  выходного нейрона.

Введем обозначения:

$M$  – количество скрытых нейронов,

$N$  – количество выходных нейронов,  
 $y_i$  – выход  $i$ -го выходного нейрона ( $y_i = 1$  означает, что нейрон сработал, т.е. на его выходе появился спайк);  
 $g_i$  – желаемое значение выхода  $i$ -го выходного нейрона;  
 $link(h, i, l)$  – наличие связи между  $h$ -м скрытым нейроном и  $l$ -м входом  $i$ -го выходного нейрона.

Общий алгоритм обучения нейронной сети на каждом цикле ее работы выглядит следующим образом:

**For**  $i=1$  to  $N$

**If** ( $y_i = 1$  and  $g_i = 0$ ) or ( $y_i = 0$  and  $g_i = 1$ ) **Then**

$u = random()$ ;  $k_{iu} = invert(k_{iu})$ ;  $h = random() \mid link(h, i, l) \ \& \ l \neq u$ ;  
 $r = random()$ ;  $k_{ir} = invert(k_{ir})$ ;

**End\_if**

**End\_for**

Для исследования предлагаемой модели нейронной сети выбрана задача обучения избеганию столкновений некоего объекта с системой его поиска и разработан программный комплекс, обеспечивающий:

- чтение файла конфигурации нейронной сети и преобразование его во внутренние структуры нейронов и межнейронных связей;
- чтение файла конфигурации лабиринта, в котором движутся поисковая система и преследуемый им объект, формирование внутренних структур данных для них;
- симуляция поведения поисковой системы, состоящая в случайных передвижениях по лабиринту;
- симуляция обучения объекта.

Результатом обучения сети должно стать такое поведение объекта, при котором он никогда не сталкивается с системой.

Поисковая система и объект перемещаются внутри заданного лабиринта. Лабиринт представляет собой прямоугольник из стен, длина и ширина которых задается в файле конфигурации. Внутри него находится задаваемое количество прямоугольных препятствий (тоже стен) непреодолимых ни объектом, ни системой его поиска.

В процессе функционирования поисковая система случайным образом перемещается по решетке свободных от стен точек лабиринта, имеющих целочисленные координаты, в поисках объекта, который должен обучиться убегать от него. Их столкновением и выигрышем системы считается ситуация, когда система и объект оказались в соседних клетках решетки.

Объект способен «видеть» состояние некоторых соседних по отношению к его текущему местоположению точек решетки. Для этого в программной модели сети предусмотрены два слоя входных нейронов.

Один слой «лоцирует» препятствия, другой – поисковую систему. Нейроны входных слоев по дендритам являются потенциальными, по аксону (выходу) – импульсными.

Область видимости точек решетки нейронами входного слоя вначале была определена так, как показано на рис. 1.a. Веса импульсов при обнаружении нейроном препятствия (или поисковой системы) установлены обратно пропорционально расстоянию до него и равны 8, 4, 2, 1 для дистанций 1, 2, 3, 4.

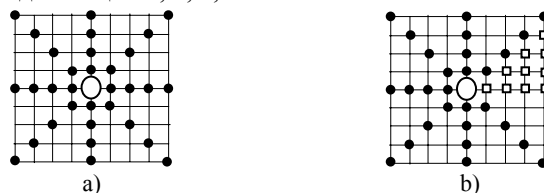


Рис. 4 Покрывтие видимости входными нейронами точек решетки, соседствующих с местоположением объекта

Однако при выполнении первых же экспериментов по симуляции сети выяснилось, что «дырчатая» область видимости сильно уменьшает способность объекта к обучению, поскольку поисковая система "внезапно" может оказаться на расстоянии 2 от него и быть «невидимой». Поэтому для каждого входного нейрона этого слоя координаты «лоцируемых» им точек были расширены путем добавления всех точек, находящихся на сетке между его направлением и ближайшим направлением против часовой стрелки. На рис. 1.b. точки, «лоцируемые» нейроном с номером 0 показаны не кружками, а квадратами. Для всех остальных нейронов соблюдается этот же принцип.

Веса входов нейронов теперь задаются так: 8, 4, 4, 2, 2, 2, 1, 1, 1, 1.

Существует скрытый слой S-нейронов, предназначенный для преобразования информации, получаемой входными слоями в последовательности импульсов, обрабатываемые нейронами выходного слоя. Каждый нейрон скрытого слоя связан по входу с каждым входным нейроном, но в выработке его выходного спайка участвуют только импульсы с тех входов, которым соответствуют единичные значения ключа.

Нейроны выходного слоя предназначены для выработки сигналов, управляющих движением объекта. Их четыре, соответственно четырем возможным направлениям перемещения, образующим пары: «вперед – назад» и «влево – вправо». Перемещение осуществляется, если сработал только один из нейронов пары и только в том случае, если этому не мешает препятствие лабиринта. Если сработали по одному нейрону каждой пары, то объект перемещается по соответствующей диагонали.

По входам каждый нейрон выходного слоя связан со всеми нейронами скрытого слоя.

Алгоритм обучения в целом соответствует тому, который описан выше, но есть некоторое отличие. Прежде всего, отметим, что симулятор помнит, ключ какого выходного нейрона модифицировался в последний раз и в какой позиции. На каждом шаге симуляции сети после выполнения перемещений системы и объекта оценивается, как изменилось расстояние между ними. Если расстояние не уменьшилось, то никаких изменений в значения ключей не вносится. В противоположном случае определяется, какой из выходных нейронов «ответственен» за перемещение объекта ближе к поисковой системе. Если этот нейрон не модифицировался на предыдущем шаге обучения, то случайным образом выбирается и инвертируется один разряд его ключа. В противном случае выбирается произвольный нейрон скрытого слоя, связь с которым не затрагивалась на предыдущем шаге обучения данного выходного нейрона и случайным образом инвертируется один разряд его ключа. Процесс продолжается либо до столкновения системы и объекта, либо по истечении заданного симулятору количества шагов.

Для того, чтобы проверить обучаемость предложенной модели, была проведена серия экспериментов. В качестве критерия для оценки обучаемости было выбрано время между очередными столкновениями системы со случайно движущимся объектом - коллизий.

Результаты двух экспериментов с KTSNN (с обучением и без обучения) показаны в таблице. Здесь: NC – количество коллизий за время моделирования  $t$ , AT – среднее время между коллизиями.

Таблица 1. – Результаты экспериментов.

	<b>Последовательность интервалов времени между коллизиями</b>	<b>NC</b>	<b>AT</b>
Без обучения ( $t=19261$ )	718, 1014, 36, 942, 1764, 15, 720, 3033, 1621, 2126, 802, 27, 80, 15, 2589, 112, 153, 51, 20, 1735, 97, 1591	22	875.5
С обучением ( $t=18312$ )	756, 539, 396, 416, 29, 44, 19, 117, 2460, 8579, 240, 2108, 58, 112, 2439	15	1220.8

Результаты проведенных экспериментов показывают, что обучение нейронной сети KTSNN существенно снижает количество столкновений и увеличивает время между очередными столкновениями.

Предложенная модель нейронной сети KTSNN (Key-Threshold Spiking Neural Network) позволяет создавать «ориентированные на

аппаратную реализацию» структуры для решения задач в реальном времени, например, навигации в робототехнике.

Эта модель нейронной сети может быть также использована для реализации нейронных сетей резервуарного типа LSM (Liquid State Machine) [3]. В модели LSM резервуар состоит из спайковой нейронной сети, обычно реализуемой на базе так называемых LIF-нейронов. Такая сеть не обучается в процессе работы LSM, а создается в процессе инициации сети LSM. Представляется, что предложенная модель является более подходящей, т.к. требует меньшего количества нейронов при той же функциональности. Планируются эксперименты с KTSNN и в рамках модели LSM.

Модель KTSNN также планируется использовать для построения самообучаемой сети, предложенной в [4].

#### **Литература:**

1. Gerstner W, Kistler W. M. Spiking Neuron Models. Single Neurons, Populations, Plasticity. - Cambridge University Press, 2002.
2. Gavrilov A. V. An Architecture of Neurocomputer for Image Recognition. // Neural Network World/ - 1991. - N.1. - P. 59-60.
3. Zhang Yanduo, Wang Kun. The application of Liquid State Machines in robot path planning. // Journal of Computers. – 2009. - Vol. 4, N. 11. - P. 1183-1186.
4. Малявко А.А., Гаврилов А.В. К вопросу о создании самообучающейся и самомодифицирующейся импульсной нейронной сети в качестве модели мозга. // Актуальные проблемы электронного приборостроения АПЭП-2016: труды XIII межд. научно-техн. конф., Новосибирск. – Новосибирск: изд-во НГТУ. - Том 9. - С. 66-69.

#### **Spiking neural network based on “key-threshold” model**

**Maliavko A.A.<sup>1</sup>, Gavrilov A.V.<sup>1</sup>**

Novosibirsk State Technical University<sup>1</sup>

*A novel model of Key-Threshold based Spiking Neural Network (KTSNN) is proposed. This neural network consists of quasi-neurons oriented to recognize any key - spikes distributed in time (sequence of spikes) or in space (in synapses). Software implementation of this model is suggested. Possible methods of learning, implementation and usage of this model are discussed..*

**Keywords: neural network, spiking neuron, machine learning.**