

## Импульсные нейронные сети.

Последние достижения в области искусственного интеллекта и машинного обучения затронули различные аспекты человеческой жизни. С появлением беспилотных автомобилей, распространением виртуальных помощников и высокоинтеллектуальных поисковых систем мы ощущаем присутствие искусственного интеллекта в нашей жизни больше, чем когда-либо. Термин «искусственная нейронная сеть» или просто «нейронная сеть» возник по аналогии с биологическими вычислительными единицами в мозге, известными как нейроны. Нейронные сети – это попытка смоделировать процессы обработки информации мозгом. Они используются в качестве мощных вычислительных инструментов для решения задач распознавания образов, прогнозирования, управления, классификации и других задач.

Первое поколение нейросетей определяется понятием перцептрона, разработанного Ф. Розенблаттом [1] в 1958 году. Простой перцептрон состоит из трёх типов элементов: S-элементов, A-элементов и одного R-элемента. S-элементы соединяются с A-элементами с помощью тормозных и возбуждающих связей, которые соответствуют весам -1 и 1 соответственно. A-элемент возбуждается только тогда, когда сумма сигналов, пришедших к нему от S-элементов, превысила некоторую фиксированную величину. Сигналы от возбужденных A-элементов, в свою очередь, передаются R-элементу. Причём сигнал от  $i$ -го A-элемента передаётся с коэффициентом  $w_i$ , который называется весом. Так же как и A-элементы, R-элемент подсчитывает сумму значений входных сигналов, умноженных на веса  $w_i$ , и сравнивает ее с порогом  $\theta$ . Перцептрон выдаёт 1, если линейная форма от входных сигналов превышает порог  $\theta$ , 0 иначе. Математически, функцию, реализуемую R-элементом, можно записать так:

$$f(x) = \begin{cases} 1, & \sum_{i=1}^n w_i x_i + \theta > 0 \\ 0, & \text{иначе} \end{cases} \text{ – функция активации,} \quad (1)$$

где  $x_i$  – входящий сигнал от  $i$ -го A-элемента, а  $w_i$  – соответствующий ему вес. Важной особенностью этих нейронных сетей является то, что функция  $f(x)$  принимает только бинарные значения. Поэтому такие нейронные сети являются универсальными для вычислений с бинарными входными и выходными данными, а с помощью многослойного перцептрона с одним скрытым слоем может быть вычислена любая булева функция.

Для эффективной работы любую нейронную сеть необходимо обучить. В процессе обучения необходимо найти такие значения весов между нейронами, чтобы уменьшить разницу между заданными векторами и векторами, получаемыми на выходе нейросети. Обучение перцептрона может происходить с учителем и без учителя с помощью метода коррекции ошибки, метода обратного распространения ошибки, а также с помощью альфа-системы подкрепления.

Улучшение понимания обработки информации мозгом привело к возникновению более сложных моделей нейронных сетей. Ко второму поколению относятся нейросети, в которых используется непрерывная функция активации, часто сигмоида или радиально-базисная функция. Такая функция активации принимает вещественные значения вместо фиксированного порогового значения в (1). Поэтому нейросети второго поколения подходят для вычислений с вещественными входными и выходными данными. Примерами нейросетей второго поколения являются многослойные нейронные сети с прямой связью, рекуррентные нейронные сети, а также нейронные сети с радиальной базисной функцией. Обучение может производиться с учителем и без учителя. К основным правилам и методам относятся правило Хейбба, метод обратного распространения ошибки и его различные модификации, а также правило "победитель получает всё".

Нейронные сети второго поколения являются отличным инструментом для решения нелинейных задач. Они обладают важным свойством, что любую нелинейную функцию можно аппроксимировать с произвольной точностью [6], [7], [13]. Также нейронные сети второго поколения используются для решения задач распознавания образов, оценки функций и задач классификации. Однако, в сравнении с человеческой способностью, вычислительная производительность нейрона еще не достигла своего полного потенциала, поскольку временная информация не была учтена. Это привело к необходимости создания биологически более правдоподобных моделей нейронных сетей, известных, как импульсные.

Импульсные нейронные сети относятся к третьему поколению. Они обладают большей вычислительной мощностью для моделирования различных нейронных сигналов [17] и, как и нейронные сети второго поколения, аппроксимируют любую непрерывную функцию [16]. Можно выделить два подхода к построению таких сетей. Первый – это подробные математические модели импульсного нейрона, основанные на детальном моделировании мембранного потенциала нейрона и проводимости ионных токов. Такие модели оказались полезными для изучения поведения биологических нейронных сетей, а также механизмов нейронного обучения, таких как

долговременная потенция и передача сигналов на основе нейромедиаторов. Однако уровень детализации этих моделей хотя и идеален для точного воспроизведения электрофизиологических реакций, увеличивает сложность модели, что затрудняет её анализ и налагает значительную вычислительную нагрузку. Второй тип подхода – это модели импульсного отклика (Spike Response Model). В отличие от подробных математических моделей, выраженных в виде дифференциальных уравнений, модели импульсного отклика имеют вид аналитического выражения с функциями сходства. Модели импульсного отклика предпочтительнее для системных исследований памяти, нейронного кодирования и сетевой динамики. В обоих типах моделей импульсный нейрон имеет неотъемлемую динамическую природу, которая характеризуется его внутренним состоянием, изменяющимся во времени. Когда внутреннее состояние нейрона достигает некоторого порога, нейрон срабатывает и генерирует потенциал действия или импульс, который передается другим нейронам. Поэтому в импульсных нейронных сетях информация передается с помощью последовательности импульсов.

В данной лекции мы рассмотрим модели импульсной нейронной сети, как первого типа, так и второго, а также методы их обучения.

## 1 Нейронная модель импульсного отклика.

**Модель импульсного отклика** была введена W. Gerstner [8]. Архитектура модели импульсного отклика похожа на модель нейронной сети с прямой связью, то есть каждый нейрон любого слоя связан с нейронами предыдущего слоя и связи между нейронами не образуют циклы. Информация движется только в одном направлении: от входящих нейронов к выходящим. В отличие от нейронной сети с прямой связью, где два нейрона связаны только одним синапсом, в модели импульсного отклика связь между двумя нейронами моделируется несколькими синапсами. Число синапсов между двумя любыми нейронами фиксированно, а каждый синапс имеет вес и задержку, ассоциированную с ним. Когда внутреннее состояние нейрона или другому мембранный потенциал пересекает некоторый порог  $\theta$ , нейрон генерирует потенциал действия или импульс. На следующем рисунке показана архитектура простейшей модели импульсного отклика.

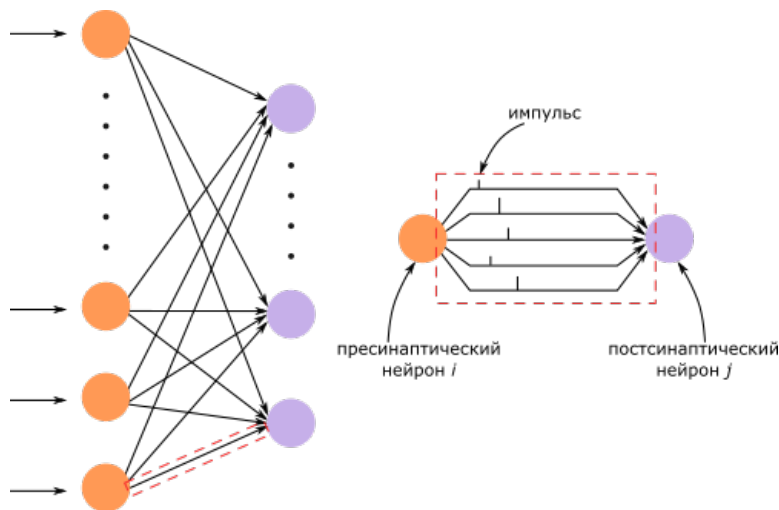


Рис. 1: Архитектура модели импульсного отклика. Каждый пресинаптический нейрон  $i$  связан с постсинаптическим нейроном  $j$  несколькими синапсами со своими весами  $\omega_{ij}$  и задержками  $d^k$ .

Пусть нейрон  $j$  связан с множеством  $\Gamma_j$  пресинаптических нейронов. Предположим, что пресинаптический нейрон  $i$  посылает импульс в момент  $t_i$ . Этот импульс достигает синапса между  $i$ -ым пресинаптическим нейроном и  $j$ -ым постсинаптическим нейроном в момент  $t_i + d^k$ , где  $d^k$  – задержка или время передачи, ассоциированная с  $k$ -ым синапсом. Динамика внутреннего состояния нейрона  $j$  описывается суммой функций импульсного отклика  $\varepsilon$ , умноженных на веса:

$$x_j(t) = \sum_{i \in \Gamma_j} \sum_{k=1}^m \omega_{ij}^k \varepsilon(t - t_i - d^k), \quad (2)$$

где  $\omega_{ij}^k$  – синаптический вес  $k$ -го синапса. Функция импульсного отклика  $\varepsilon$  описывает влияние пресинаптиче-

ского импульса на внутреннее состояние постсинаптического нейрона. Она имеет вид:

$$\varepsilon(t) = \begin{cases} \frac{t}{\tau} e^{1-\frac{t}{\tau}}, & t > 0 \\ 0, & t \leq 0 \end{cases}$$

где  $\tau > 0$  – постоянная, которая определяет время нарастания и затухания внутреннего состояния постсинаптического нейрона. Функция  $\varepsilon$  принимает максимальное значение 1 в точке  $t = \tau$ .

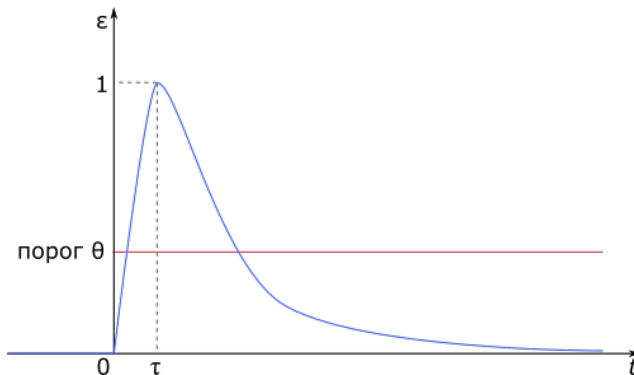


Рис. 2: Функция импульсного отклика.

Для простоты нейроны в данной модели ограничены передачей одного импульса в каждом синапсе. Также существует модели, в которых информация от одного нейрона к другому передается с помощью нескольких импульсов в каждом синапсе [3], [11].

Импульсные нейронные сети отличаются от нейронных сетей предыдущих поколений тем, что информация, передается с помощью последовательности импульсов. Следовательно, входные данные, которые обычно имеют другую форму (аналоговую), необходимо преобразовать в дискретные временные данные импульсов.

Все схемы кодирования можно разделить на две основные категории: кодирование скоростью и временное кодирование. Популяционное кодирование, которое часто относят к третьей категории, только добавляет информацию о количестве нейронов в используемой схеме кодирования.

Кодирование скоростью предполагает, что информация о сигнале содержится в скорости срабатывания нейрона. На практике скорость измеряется усреднением количества импульсов в определенном интервале времени или усреднением количества импульсов по нескольким повторениям эксперимента. Хотя схема кодирования усреднением количества импульсов в определенном интервале времени широко используется в различных нейронных сетях, к реальной работе мозга она не имеет отношения, так как она зависит от выбора интервала времени. Чтобы получить разумные средние значения во временном интервале должны произойти несколько импульсов (подробнее в [9]). Поэтому такая схема кодирования подходит в случае постоянных или медленно меняющихся стимулов. Однако, входные данные в реальном мире меняются быстро. Например, изображение, проецируемое на фоторецепторы сетчатки человека, меняется каждые несколько сотен миллисекунд. Схема кодирования усреднением количества импульсов по нескольким повторениям эксперимента также не является биологически правдоподобной схемой кодирования, используемой нейронами в мозге, так как нейроны не могут ждать повторения стимулов одинаковым образом в каждом эксперименте, прежде чем сгенерировать ответ (подробнее в [10]). В качестве примера возьмем лягушку, которая хочет съесть муху. Лягушка не ждет пока муха пролетит мимо несколько раз по одному и тому же маршруту, чтобы принять решение. Она принимает решение, основываясь на одном пролете, так как маршруты пролёта мухи будут отличаться.

Временное кодирование предполагает, что информация, передаваемая нейронами, содержится в точном времени импульсов и интервалах между импульсами. Временное кодирование использует те свойства импульсной активности, которые не могут быть описаны кодированием скоростью. Например, для очень коротких стимулов максимальная скорость возбуждения нейрона может быть недостаточно высокой, чтобы произвести более одного импульса. Поэтому время возникновения импульса может нести более полезную информацию. Существует много схем, которые относятся к временному кодированию [2]. Самой простой схемой является кодирование с помощью времени первого импульса (TTFs), где информация кодируется с помощью временной разницы между стимулом и первым импульсом, который сгенерирован нейроном. Еще одна схема,

относящаяся к временному кодированию, – это фазовое кодирование. Фазовое кодирование информации применяется в случае периодического сигнала. Известно, что в гиппокампе, в обонятельной системе, а также в других областях мозга колебания нейронов происходят постоянно. Эти колебания могут служить внутренним опорным сигналом. При фазовом кодировании информация кодируется с помощью временной разницы между первым импульсом и началом фазы текущих колебаний нейронной сети. Таким образом, фазовые колебания нейронной сети задают временную шкалу, относительно которой и определяется первый импульс. Экспериментально было доказано, что в некоторых кортикальных сенсорных областях коры головного мозга нейроны кодируют стимулы с помощью времени импульсов по отношению к текущим колебаниям нейронной сети.

Рассмотрим разреженное временное кодирование для описанной выше модели импульсного отклика, предложенное в работе [4]. Авторы этой работы назвали его популяционным. Разреженность означает, что для кодирования каждого элемента входящего вектора используется небольшое количество нейронов относительно всех доступных нейронов. Согласно этой схеме, кодирование входных данных происходит с помощью  $m$  нейронов, описываемых пересекающимися гауссовыми рецептивными полями. При этом каждая непрерывная компонента входящего вектора переводится во время срабатывания каждого из  $m$  нейронов. А именно, сначала для каждой  $i$ -ой компоненты входящего вектора определяется диапазон её значений  $[I_{min}^i \dots I_{max}^i]$ . Далее этот диапазон покрывается  $m$  гауссовыми рецептивными полями с центром  $c_i$  и шириной  $\sigma_i$ , определяемыми по следующим формулам:

$$c_i = I_{min} + \left( \frac{2i - 3}{2} \right) \left( \frac{I_{max} - I_{min}}{m - 2} \right),$$

$$\sigma_i = \frac{1}{\gamma} \frac{I_{max} - I_{min}}{m - 2},$$

где число  $\gamma$  выбирается экспериментально, чаще всего  $\gamma = 1.5$ . Устанавливается временной порог 9 мс. Нейрон, у которого рецептивное поле дает время срабатывания меньше этого порога, отмечается, как несработавший. На следующем рисунке изображено кодирование числа 0.3 шестью нейронами с помощью время срабатывания нейрона 1 (6.2 мс), нейрона 2 (1,3 мс), нейрона 3 (0,3 мс), нейрона 4 (3,6 мс), нейрона 5 (7,8 мс). Нейрон 6 вообще не срабатывает, так как задержка срабатывания выше установленного порога.

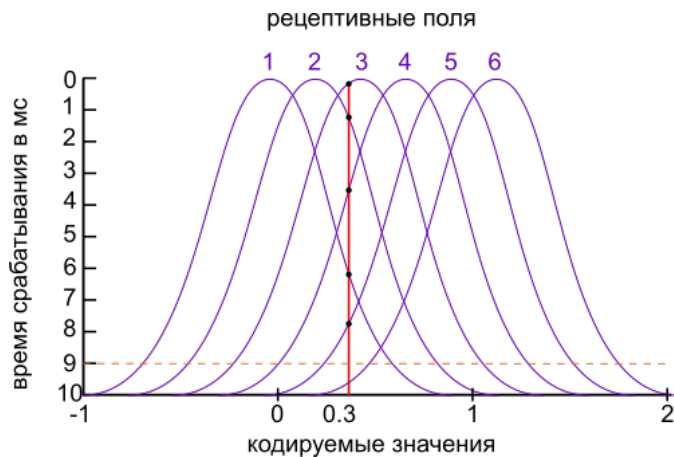


Рис. 3: Кодирование вещественного значения и время срабатывания каждого из 6 нейронов.

Экспериментально доказано, что разреженное представление сенсорной информации наблюдается во многих системах, включая зрение, слух, осязание и обоняние. Например, в нашей Лекции 2 мы рассматривали схему обоняния фруктовой мухи, где используется разреженное кодирование запахов.

## 2 Обучение нейронной модели импульсного отклика.

Для нейронной модели импульсного отклика существуют как алгоритмы по обучению с учителем, так и без учителя. Первым алгоритмом, который выполнял обучение с учителем, был SpikeProp [5], основанный на методе обратного распространения ошибки. В качестве ошибки использовалась разница между ожидаемым и реальным временем появления импульса. При этом предполагалось, что нейрон может передавать

только один импульс за один цикл. Это ограничение помогало обойти отсутствие непрерывности функции мембранного потенциала в моменты срабатывания нейрона. В связи с этим применение данного алгоритма для решения более сложных задач ограничено. В дальнейшем для обучения с учителем нейронной модели импульсного отклика применялись алгоритмы QuickProp, RProp, Levenberg-Marquardt BP, которые повышали вычислительную эффективность. А в [3], [11] предложены алгоритмы, где предполагается, что нейроны могут передавать несколько импульсов за один цикл. Помимо алгоритмов, основанных на методе обратного распространения ошибки, отметим еще алгоритм, основанный на статистическом подходе [20].

Алгоритмы по обучению без учителя нейронной модели импульсного отклика часто используют различные варианты правила Хебба и соревновательное обучение для модификации синаптических весов. Рассмотрим обучающий алгоритм, предложенный в [19]. В данной работе правило "победитель получает все" используется для корректировки весов между пресинаптическими нейронами и первым сработавшим постсинаптическим нейроном, то есть внутреннее состояние (или мембранный потенциал) которого пересекло порог после поступления в синапс импульсов от всех пресинаптических нейронов. Тогда веса между пресинаптическим нейроном  $i$  и сработавшим постсинаптическим нейроном  $j$  обновляются согласно правилу Хебба. А именно, если постсинаптический нейрон генерирует импульс сразу после получения импульса на синапс от пресинаптического нейрона, тогда синаптический вес между этими нейронами увеличивается. Если импульс постсинаптического нейрона сгенерирован до получения импульса от пресинаптического нейрона, тогда вес между соответствующими нейронами уменьшается. Более точно, веса  $\omega_{ij}^k$  между пресинаптическим нейроном  $i$  и постсинаптическим нейроном  $j$  с задержками  $d^k$  обновляются согласно следующему правилу:

$$\Delta\omega_{ij}^k = \eta L(\Delta t_{ij}), \quad (3)$$

где  $\eta > 0$  – скорость обучения,  $L(\Delta t_{ij})$  – обучающая функция,  $\Delta t_{ij} = t_{pre} - t_{post}$  – разница между временем прибытия импульса пресинаптического нейрона и постсинаптического нейрона в синапс. Импульс, выпущенные из пресинаптического нейрона  $i$  в момент времени  $t_i$ , достигает синапса в момент  $t_{pre} = t_i + d^k$ . Импульс, выпущенные из постсинаптического нейрона  $j$  назад в синапс в момент времени  $t_j$ , достигает синапса в момент  $t_{post} = t_j + d^{back}$ . Подробнее на следующем рисунке.

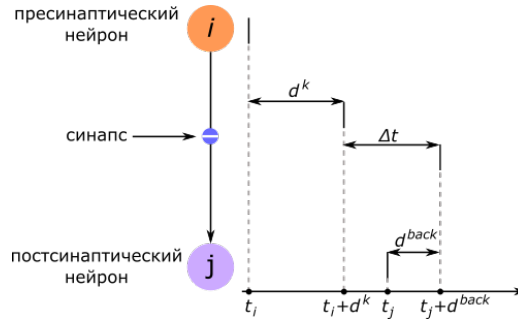


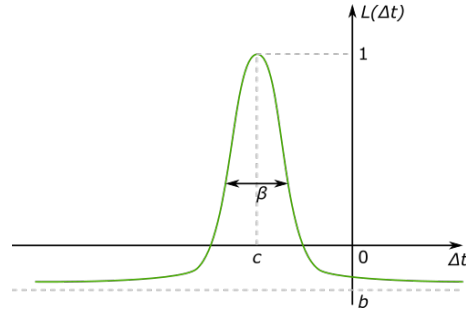
Рис. 4: Как только импульс от пресинаптического нейрона  $i$  поступил на синапс и внутренне состояние постсинаптического нейрона  $j$  пересекло порог, нейрон  $j$  генерирует импульс и посылает его назад в синапс.

Веса  $\omega_{ij}^k$  меняются от нуля до максимального значения  $\omega^{max}$ . Изначально всем весам  $\omega_{ij}^k$  присваиваются случайные значения такие, что постсинаптический нейрон не может сработать, если не получил хотя бы один импульс от каждого пресинаптического нейрона. В качестве обучающей функции  $L(\Delta t)$  в (3) рассматривается Гауссова кривая

$$L(\Delta t) = (1 - b)e^{-\frac{(\Delta t - c)^2}{\beta^2}} + b,$$

где  $\beta > 0$ ,  $b < 0$ ,  $c < 0$  – параметры. Параметр  $c$  определяет позицию пика функции  $L(\Delta t)$ , обычно его выбирают между 2 и 3 миллисекундами,  $\beta$  влияет на ширину "колокола" функции. Выбор параметра  $b$  имеет критическое значение, так как может приводить к абсолютно разным результатам в процессе обучения. Чаще всего  $b$  выбирается в числовом интервале  $(-0.3, 0)$ .

Выбор обучающей функции в виде Гауссовой кривой согласуется с правилом Хебба. На следующем графике видно, что если  $\Delta t$  принимает значения внутри отрезка, который соответствует положительной части "колокола", тогда веса  $\omega_{ij}^k$  между пресинаптическими нейронами и постсинаптическим нейроном-победителем увеличиваются, так как  $\Delta\omega_{ij}^k = \omega_{ij}^{k,new} - \omega_{ij}^{k,old} = \eta L(\Delta t_{ij})$ . Если  $\Delta t$  принимает значения на всей остальной части оси, то веса между соответствующими нейронами будут уменьшаться.

Рис. 5: График обучающей функции  $L(\Delta t)$ .

В работе [19] для кодирования входных данных использовалось обычное (не разреженное) временное кодирование. При применении данного метода обучения к реалистичным данным возникали проблемы. А именно, нельзя было успешно кластеризовать данные, содержащие существенно больше кластеров, чем размерность входных данных, особенно в случае низкой размерности. Эта проблема связана с минимальной шириной  $\beta$  обучающей функции  $L(\Delta t)$ , влияющей на пространственный размер кластера, полученного в процессе обучения. Например, для двумерных входных данных, содержащих более, чем два кластера, описанный выше алгоритм обучения не работал для большинства значений параметров. Кроме того, конечная ширина обучающей функции препятствует обнаружению нескольких соседних кластеров меньшего размера относительно ширины обучающей функции. Поэтому было разработано разреженное временное кодирование входных данных с помощью рецептивных полей [4], описанное выше. Присваивая время срабатывания только небольшому числу активных нейронов, мы получаем разреженное кодирование, позволяющего обучать только активные синапсы. Такое кодирование повышает мощность и точность обучающего алгоритма.

### 3 Нейронная модель Ижикевича.

Наиболее реалистичная модель нейрона создана Аланом Ллойдом Ходжкиным и Эндрю Хаксли в 1952 году [12] для описания электрических механизмов, которые лежат в основе генерации и передаче нервного сигнала в гигантском аксоне кальмара. Однако модель Ходжкина-Хаксли сложна как в вычислительном отношении, так и в плане анализа её динамики. Евгений Ижикевич создал упрощенную модель нейрона на основе уравнений Ходжкина-Хаксли [15]. Эта модель проста в вычислительном отношении. Она описывает изменение электрического потенциала в мембране нейрона в зависимости от тока, протекающего через ионные каналы мембраны. Эти изменения электрического потенциала выражаются в виде двух дифференциальных уравнений следующим образом:

$$\frac{dv}{dt} = 0.04v^2 + 5v + 140 - u + I, \quad (4)$$

$$\frac{du}{dt} = a(bv - u), \quad (5)$$

со вспомогательным сбросом

$$\text{если } v \geq 30 \text{ mV, то } \begin{cases} v \leftarrow c \\ u \leftarrow u + d, \end{cases} \quad (6)$$

где  $v$  представляет мембранный потенциал нейрона,  $u$  – вспомогательная переменная, восстанавливающая мембранный потенциал, которая отвечает за активацию ионных токов  $K^+$  и инактивацию ионных токов  $Na^+$ , и обеспечивает отрицательную обратную связь с  $v$ . Когда мембранный потенциал  $v$  увеличивается, восстанавливающая переменная  $u$  уменьшается. Также  $u$  влияет на время между импульсами. После того, как импульс достигает своего пикового значения  $30 \text{ mV}$ , мембранный потенциал и вспомогательная переменная сбрасываются в соответствии с (6). Потенциал покоя в модели составляет от  $70$  до  $60 \text{ mV}$  в зависимости от значения параметра  $b$ . Порог  $30 \text{ mV}$  также не фиксированный, он может быть, как больше, так и меньше, в зависимости от истории поведения мембранного потенциала по отношению к импульсу.  $I$  – внешний постоянный ток или синаптические токи,  $a, b, c, d$  – параметры, путём настройки которых модель (4)–(6) может воспроизводить почти все виды кортикальных нейронов: **regular spiking neurons** – нейроны запускают

несколько импульсов с коротким периодом между ними, который постепенно увеличивается, **intrinsically bursting neurons** – нейроны запускают вспышку или взрыв импульсов, за которой следуют повторяющиеся одиночные импульсы, **chattering neurons** – нейроны запускают вспышки близко расположенных импульсов, частота между импульсами внутри вспышки может достигать 40 Гц, **fast spiking neurons** – нейроны очень быстро запускают импульсы с высокой частотой без какого-либо замедления и многие другие. На следующем рисунке примеры перечисленных нейронов.

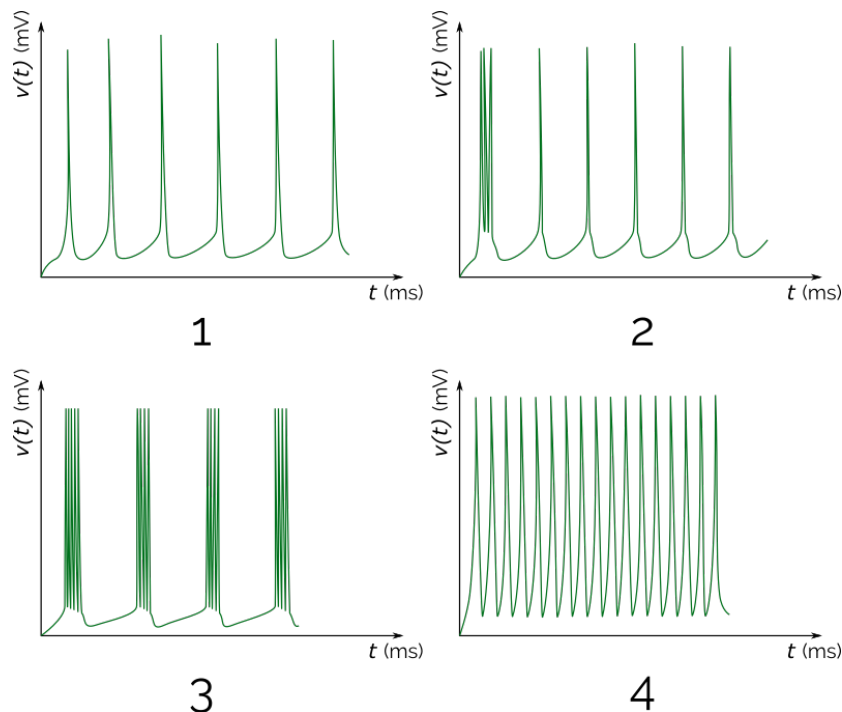


Рис. 6: Различные типы нейронных ответов: 1 – regular spiking neurons, 2 – intrinsically bursting neurons, 3 – chattering neurons, 4 – fast spiking neurons.

Рассмотрим, как параметры  $a$ ,  $b$ ,  $c$  и  $d$  влияют на  $v$  и  $u$ . Параметр  $c$  является значением, которое принимает мембранный потенциал  $v$  после возникновения импульса. Параметр  $d$  влияет на время между импульсами мембранного потенциала. А именно, после того, как возник импульс, происходит сброс, согласно (6), и наступает период восстановления мембранного потенциала  $v$  до следующего импульса. При этом восстанавливающаяся переменная  $u$  после возникновения импульса увеличивается путем добавления параметра  $d$ . Чем меньше значение  $d$ , тем короче период восстановления мембранного потенциала  $v$ . Параметр  $d$  всегда положителен, иначе это означало бы, что период восстановления постоянно уменьшается, автоматически увеличивая  $v$ , и импульсы происходили бы постоянно. Параметр  $a$  влияет на скорость восстановления вспомогательной переменной  $u$  после сброса. Чем меньше значение  $a$ , тем медленней происходит восстановление  $u$ . Параметр  $b$  влияет на чувствительность  $u$  к флуктуациям  $v$ . Чем больше значение  $b$ , тем сильнее связь между  $u$  и  $v$ , что приводит к подпороговым колебаниям (колебания, которые не достигают порога) между импульсами и низкопороговой динамики импульсов.

## 4 Реализация нейронной модели Ижикевича в задачах распознавания.

Существует немного работ, посвященных реализации нейронной модели Ижикевича. Например, в работах [23], [24] показано, как один нейрон Ижикевича можно применить для линейных и нелинейных задач распознавания образов. Идея состоит в использовании частоты срабатывания нейрона. Сначала каждый входящий образ умножается на весовой вектор, чтобы получить внешний ток. Затем нейрон Ижикевича стимулируется в течение  $T$  мс и вычисляется частота срабатывания. После регулировки синаптических весов входящие образы, принадлежащие к одному классу, будут генерировать почти одинаковую частоту срабатывания, а входящие образы, принадлежащие разным классам, будут генерировать достаточно разную частоту срабатывания.

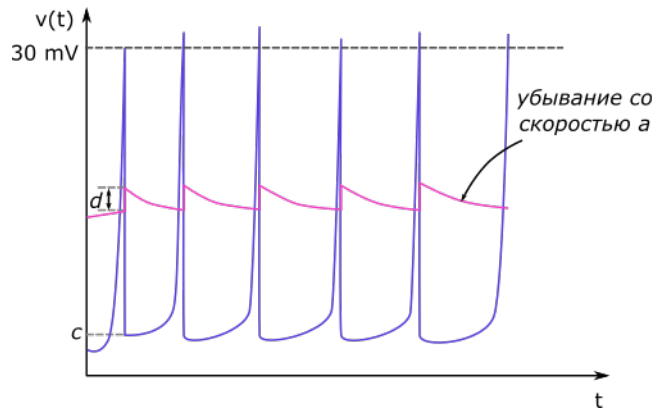


Рис. 7: Изменение мембранного потенциала  $v$  (фиолетовым цветом) и восстанавливающей переменной  $u$  (малиновым цветом).

В работах [14], [21] нейронная модель Ижикевича использовалась в нейронной сети, реализованной на программируемой логической матрице FPGA, для распознавания изображений букв или цифр. Нейронная сеть в данных работах состояла из двух слоев. Каждый нейрон первого слоя соответствует конкретному пикселю в изображении, подаваемому на вход. Поэтому количество нейронов первого слоя равно количеству пикселей во входящем изображении. Для нейронов первого слоя внешний ток  $I$  равен нулю, если пиксель, соответствующий данному нейрону, "выключен". Если пиксель "включён", то внешний ток  $I$  равен постоянному значению. Число нейронов второго слоя равно числу изображений, представленных нейронной сети при обучении. Каждый нейрон первого слоя связан с каждым нейроном второго слоя. Полный ток для каждого нейрона второго слоя равен сумме токов, полученных от каждого нейрона первого слоя. А ток от каждого нейрона первого слоя равен либо весу между соответствующими нейронами, если нейрон первого слоя сработал, либо нулю, если нейрон первого слоя не сработал. Элементы весовой матрицы определяются в процессе обучения. На этапе распознавания изображение подается на первый уровень нейросети. После определенного количества циклов срабатывает один нейрон второго слоя, идентифицируя входное изображение.

В нашей лекции мы рассмотрим нейронную сеть, состоящую из нейронов Ижикевича, для распознавания произнесенных голосом чисел от нуля до 9 [22]. А именно, рассматривается следующая система дифференциальных уравнений:

$$100 \frac{dv}{dt} = 0.74v^2 + 70v + 1680 - u + I, \quad (7)$$

$$\frac{du}{dt} = a(bv - u - 120), \quad (8)$$

со вспомогательным сбросом

$$\text{если } v \geq 35 \text{ mV, то } \begin{cases} v \leftarrow c \\ u \leftarrow u + d, \end{cases} \quad (9)$$

где параметры  $a = 0.03$ ,  $b = -2$ ,  $c = -50$ ,  $d = 100$  определяют вид нейрона, имеющего регулярные импульсы (рисунок 6 под номером 1).

Перед тем, как данные будут подаваться на вход нейронной сети, сигнал необходимо преобразовать. Речевой сигнал делится на небольшие перекрывающиеся временные отрезки, называемые речевыми кадрами или фреймами. Для каждой произнесенной цифры требуется фиксированное число фреймов  $N = 40$ . Так как длина  $L$  произнесенной цифры варьируется от 500 до 1000 мс, мы разделим эту длину на  $N$  фреймов с 50% перекрытием, чтобы поддержать длину фрейма 40-50 мс. За 50% перекрытие отвечает величина  $\gamma = 0.5$ . То есть длина фрейма вычисляется на основе  $N$ ,  $L$  и  $\gamma$  по следующей формуле:

$$\text{длина фрейма (мс)} = \frac{L(\text{мс})}{N(1 - \gamma) + \gamma}.$$

После деления звукового сигнала на фреймы из каждого фрейма выделяется вектор признаков. Часто для этого используют метод мел-кепстральных коэффициентов (описывался в нашей Лекции 1). В рассматриваемой нами работе [22] это делается следующим образом. Сначала вычислим частотный спектр для каждого



фрейма:

$$\text{частотный спектр} = \log |FFT(\text{фрейм})|^2,$$

где  $FFT$  – быстрое преобразование Фурье. На следующем рисунке представлена спектрограмма произнесенного слова *seven*.

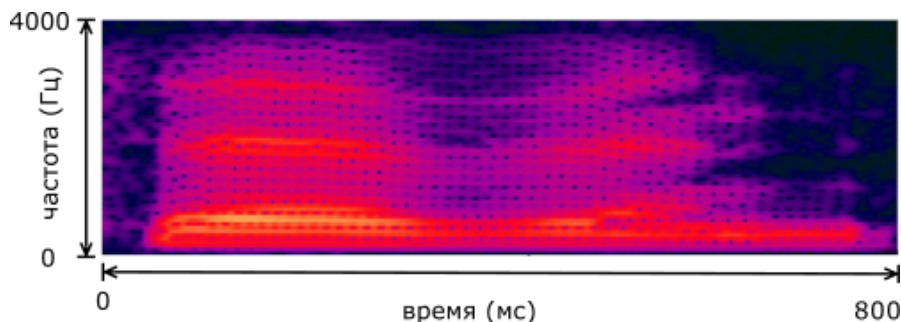


Рис. 8: Спектрограмма произнесенного слова *seven*. Цветом показана энергия: желтым – самая большая, синим – самая маленькая.

Низкие частоты в спектрограмме обладают большей энергией, чем высокие, и содержат информацию, относящуюся к классификации. Поэтому создаются частотные диапазоны. Фрейм, охватывающий  $R = 4000$  Гц, делим на  $M = 5$  частотных диапазонов. На каждом из пяти диапазонов вычисляется средняя энергия. Значение каждого элемента вектора признаков – это и есть средняя энергия в каждом диапазоне. Если каждый фрейм поделить на большее число диапазонов, то это увеличит число элементов в векторе признаков и увеличит вычислительную нагрузку нейросети. Для определения диапазонов используется последовательность чисел Фибоначчи, которая создаёт подходящие размеры частотных диапазонов для маленького числа элементов вектора признаков. После деления получаем частотные диапазоны с длинами 333,3, 333,3, 666,7, 1000, и 1666,7 Гц. Для определения диапазонов не обязательно использовать числа Фибоначчи, возможны и другие подходы.

Архитектура нейронной сети изображена на следующей картинке.

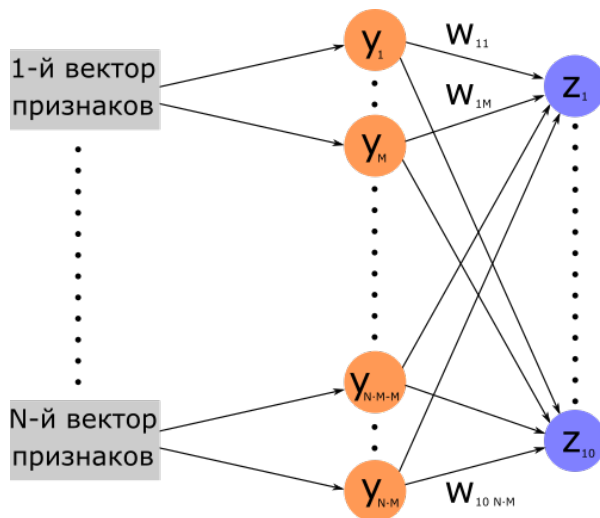


Рис. 9: Архитектура нейронной сети.

На вход нейронной сети подаются  $N$  векторов признаков. Каждый вектор признаков имеет  $M = 5$  компонент. Каждая компонента вектора признаков становится внешним током для нейронов  $y_j$ , которые реализуются, как нейроны Ижикевича (7)–(9). Общее число нейронов  $y_j$  равно  $N \cdot M = 200$ . Выходной слой состоит из нейронов  $z_k$ , которые также реализованы, как нейроны Ижикевича (7)–(9). Каждый нейрон  $z_k$  имеет синаптический ток из 200 нейронов  $y_j$ . Всего в выходном слое 10 нейронов  $z_k$ , каждый из которых соответствует одному из 10 классов произнесенных чисел. Опишем подробнее синаптический ток для нейронов  $z_k$ .

Обозначим с помощью  $G_{kj}(t)$  функцию, определяющую синаптическую проводимость  $j$ -го синапса нейрона  $z_k$ , изменяющуюся во времени при получении одного входящего импульса в этот синапс. Пусть она имеет вид:

$$G_{kj}(t) = K_{kj}(t - t_j)e^{-\frac{t-t_j}{\tau}}, \quad j = 1, \dots, N \cdot M, \quad (10)$$

где константа  $K_{kj}$  управляет амплитудой проводимости,  $\tau$  – время, когда синапс достигает своей максимальной проводимости,  $t_j$  – время импульса  $j$ -го входящего синапса. Ионный ток равен проводимости, умноженной на разницу между реверсивным потенциалом и мембранным потенциалом, то есть для любого постсинаптического нейрона  $z_k$  и  $j$ -го входящего синапса

$$I_{kj} = G_{kj}(t)(E - v(t)).$$

В нашем случае реверсивный потенциал  $E = 0$ . Так как каждый нейрон  $z_k$  имеет  $N \cdot M$  входящих синапсов, то ионный ток из всех синапсов в постсинаптический нейрон  $z_k$  равен

$$I_k = \sum_{j=1}^{N \cdot M} I_{kj} = - \sum_{j=1}^{N \cdot M} G_{kj}(t)v(t). \quad (11)$$

В процессе обучения, в (11) появляется еще одна сумма произведений синаптических весов  $w_{kj}$  между пресинаптическими нейронами  $y_j$  и постсинаптическим нейроном  $z_k$ , и мембранного потенциала  $v_{pre,j}(t)$  пресинаптического нейрона  $y_j$ . Поэтому полный синаптический ток, который поступает в каждый нейрон  $z_k$

$$I = - \sum_{j=1}^{N \cdot M} G_{kj}(t)v(t) - \sum_{j=1}^{N \cdot M} w_{kj}v_{pre,j}(t).$$

Регулировка синаптических весов  $w_{kj}$  происходит согласно методу STDP (Spike Timing Dependent Plasticity) [18] с учителем. STDP может быть основан как на правиле Хебба, о котором мы говорили выше, так и на "обратном" правиле Хебба. Сигнал учителя определяет какое из STDP применяется для обновления весов. Напомним, согласно правилу Хебба, если постсинаптический нейрон генерирует импульс сразу после получения импульса на синапс от пресинаптического нейрона, тогда синаптический вес между этими нейронами увеличивается. А если импульс постсинаптического нейрона сгенерирован до получения импульса от пресинаптического нейрона, тогда вес между соответствующими нейронами уменьшается. STDP на основе этого правила Хебба имеет вид:

$$\Delta w_{kj} = \begin{cases} 0.01Ae^{-\frac{|t_k - t_j|}{\tau^+}} & t_k - t_j \geq 0, A > 0 \\ 0.01Be^{-\frac{|t_k - t_j|}{\tau^-}} & t_k - t_j < 0, B < 0 \end{cases}, \quad (12)$$

где  $t_k$  – время генерации импульса постсинаптическим нейроном  $z_k$ , а  $t_j$  – время, когда импульс пресинаптического нейрона  $y_j$  поступил на синапс между пресинаптическим и постсинаптическим нейронами,  $A, B, \tau^+, \tau^-$  – постоянные. Оба случая в (12) являются убывающими экспонентами. Чтобы получить STDP, основанное на "обратном" правиле Хебба, случаи в (12) меняются местами и получаем возрастающие экспоненты, а именно:

$$\Delta w_{kj} = \begin{cases} 0.01Be^{-\frac{|t_k - t_j|}{\tau^-}} & t_k - t_j \geq 0, B < 0 \\ 0.01Ae^{-\frac{|t_k - t_j|}{\tau^+}} & t_k - t_j < 0, A > 0 \end{cases}. \quad (13)$$

Напомним, каждый нейрон, представляющий один из 10 классов произнесенных чисел, получает 200 импульсных последовательностей длительностью 100 Мс через 200 синапсов. Во время обучения если нейрон  $z_k$  генерирует импульс, то соответствующие веса  $w_{kj}$  обновляются либо по правилу (12), либо по (13). Если  $z_k$  соответствует тому классу, звуковой сигнал которого подавался на вход нейронной сети, тогда веса между этим нейроном и двестию нейронами  $y_j$  меняются согласно правилу (12). Если нейрон  $z_k$  не соответствует тому классу, то соответствующие веса меняются по правилу (13). Перед обучением веса инициализируются случайным образом между 0 и 1, и нормализуются по норме  $\mathbb{L}_1$ .

Изменение синаптических весов влияет на изменение амплитуды проводимости в (10), а именно

$$\Delta K_{kj} = \Delta w_{kj}K_{kj}.$$

Реализацию данной нейронной сети и её обучение на языке Python можно посмотреть [здесь](#).

Импульсные нейронные сети являются естественными преемниками нейронных сетей второго поколения. Однако они пока далеки от практических инструментов для решения большинства задач. Публикации по импульсным нейронным сетям являются либо теоретическими, либо демонстрируют невысокую для современных задач производительность по сравнению с нейросетями второго поколения. Основной проблемой практического использования импульсных нейронных сетей является обучение. Известно множество методов и алгоритмов обучения импульсных нейронных сетей. Наиболее используемые из них это метод STDP и его модификации, методы на основе правила Хебба, а также методы обратного распространения ошибки SpikeProp. Но, к сожалению, все они требуют чрезмерного времени вычислений. Можно ожидать, что с развитием технологий в ближайшем будущем вычислительные ограничения можно будет преодолеть.

## Литература

- [1] Ф. Розенблатт, Принципы нейродинамики: Перцептроны и теория механизмов мозга — М.: Мир, 1965.
- [2] D. Auge, J. Hille, E. Mueller and A. Knoll, A Survey of Encoding Techniques for Signal Processing in Spiking Neural Networks, *Neural Processing Letters* 53, 2021.
- [3] O. Booi, H.T. Nguyen, A gradient descent rule for multiple spiking neurons emitting multiple spikes. *Information Processing Letters* 95(6), 2005.
- [4] S.M. Bohte, H. La Poutre, J.N. Kok, Unsupervised clustering with spiking neurons by sparse temporal coding and Multi-Layer RBF Networks, *IEEE Transactions on Neural Networks*, 13(2), 2002.
- [5] S.M. Bohte, J.N. Kok and H. La Poutre, Error-backpropagation in temporally encoded networks of spiking neurons, *Neurocomputing*, 48(1-4), 2002.
- [6] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals and Systems*, 2, 1989.
- [7] M. Garzon, F. Botelho, Dynamical approximation by recurrent neural networks. *Neurocomputing*, 29, 1999.
- [8] W. Gerstner, Time structure of the activity in neural network models, *Phys. Rev. E*, 51, 1995.
- [9] W. Gerstner, W. M. Kistler, *Spiking neuron models : single neurons, populations, plasticity*, Cambridge University Press, 2002.
- [10] W. Gerstner, W. M. Kistler, R. Naud, *Neuronal Dynamics, From Single Neurons to Networks and Models of Cognition*, Cambridge University Press, 2014.
- [11] S. Ghosh-Dastidar, H. Adeli, A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. *Neural Networks* 22(10), 2009.
- [12] A.L. Hodgkin, A.F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, *The Journal of Physiology*, 117 (4), 1952.
- [13] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators. *Neural Networks*. 2, 1989.
- [14] A. J. Humaidi, Th. M. Kadhim, S. Hasan, I. K. Ibraheem, A. Taher, A Generic Izhikevich-Modelled FPGA-Realized Architecture: A Case Study of Printed English Letter Recognition, 24th International Conference on System Theory, Control and Computing, 2020.
- [15] E. M. Izhikevich, Simple Model of Spiking Neurons, *IEEE Transactions on Neural Networks*, 14, 2003.
- [16] W. Maass, Fast sigmoidal networks via spiking neurons, *Neural Computation* 9(2), 1997c.
- [17] W. Maass, Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons, *Neural Information Processing Systems*, The MIT Press, Cambridge, MA, 1997.
- [18] H. Markram, W. Gerstner, P. J. Sjöström, A history of spike-timing- dependent plasticity, *Front. Synaptic Neuroscience*, 2011.
- [19] T. Natschläger and B. Ruf, Spatial and temporal pattern analysis via spiking neurons, *Network: Comp. Neural Syst.*, 9(3), 1998.

- [20] J.P. Pfister, D. Barber and W. Gerstner, Optimal Hebbian Learning: A Probabilistic Point of View, ICANN/ICONIP, 2714 2003.
- [21] K. L. Rice, M. A. Bhuiyan, T. M. Taha, Ch. N. Vutsinas, M. C. Smith, FPGA Implementation of Izhikevich Spiking Neural Networks for Character Recognition, International Conference on Reconfigurable Computing and FPGAs, 2009.
- [22] A. Tavanaei and A. S. Maida, A Spiking Network that Learns to Extract Spike Signatures from Speech Signals, Neurocomputing, 240, 2017.
- [23] R. A. Vazquez, Izhikevich Neuron Model and its Application in Pattern Recognition, Australian Journal of Intelligent Information Processing Systems, 11(1), 2010.
- [24] S. Valadez-Godinez, J. Gonzalez and Humberto Sossa, Efficient Pattern Recognition Using the Frequency Response of a Spiking Neuron, Mexican Conference 2017: Pattern Recognition.