

# **Обучение связей конкуренции локальных рецептивных полей в спайковых нейронных сетях**

Даниил Гафни, Вячеслав Дёмин, Дмитрий Нехаев

## **Содержание**

<b>Аннотация</b>	<b>1</b>
<b>Введение</b>	<b>2</b>
<b>1 Обучение спайковой нейронной сети с необучаемой конкуренцией локальных рецептивных полей</b>	<b>8</b>
1.1 Описание задачи . . . . .	8
1.2 Особенности архитектуры сетей с конкуренцией . . . . .	9
1.3 Обучение прямых связей . . . . .	12
1.4 Интерпретация активности спайковой нейронной сети .	15
1.4.1 Калибровка голосов . . . . .	16
1.4.2 Сравнение алгоритмов интерпретации активности сети . . . . .	18
1.5 Сравнение эффективности операции свертки и локального рецептивного поля . . . . .	21
<b>2 Обучение связей конкуренции</b>	<b>24</b>
2.1 Анализ обученных весов конкуренции . . . . .	29
<b>Выводы</b>	<b>30</b>
<b>Литература</b>	<b>32</b>

## Аннотация

Стандартные методы обучения весов связей, применяющиеся в формальных нейронных сетях (метод обратного распространения ошибки) затруднительно применять к спайковым нейронным сетям из-за их дискретной и распределенной во времени природы. Хотя такие проекты существуют, больший интерес представляют локальные алгоритмы обучения, являющиеся более биологически корректными, способными обеспечить самообучение алгоритма (без учителя) и более энергоэффективными (при условии реализации на специализированном аппаратном оборудовании).

Одним из основных недостатков СНС по сравнению с формальными НС является слабая изученность локальных правил обучения, и ввиду этого, более низкие показатели точности работы (логического вывода) алгоритмов на основе СНС. Таким образом, исследование алгоритмов обучения СНС, таких как STDP, является важной задачей.

Введение связей конкуренции (имеющих отрицательные веса) позволяет добиться лучшего разделения выучиваемых нейронами признаков. Обучение этих связей, возможно, позволит добиться еще более хороших результатов за счет организации кооперативной или конкурентной работы групп нейронов.

## Введение

Спайковые нейронные сети являются перспективным нейроморфным алгоритмом, биологически корректно моделируя взаимодействия нейронов мозга. Наибольший интерес СНС представляют для решения задач в реальном времени (принятие решений, распознавание образов), так как могут быть реализованы на специализированном вычислительно- и энергоэффективном нейрочипе (например, на основе мемристоров).

Современные формальные нейронные сети отлично справляются со многими задачами машинного обучения [1; 2]. Однако их обучение

— трудоемкий процесс, требующий больших вычислительных ресурсов. Обычно обучение ведется на десятках и сотнях тысяч примеров и может занимать месяцы. Как само обучение, так и последующее применение формальных нейросетевых алгоритмов далеки от эффективности [3]. Это связано как с физически раздельным хранением значений весов и активаций нейронов, так и с самими вычислениями, которые носят тензорный характер — связанный с векторно-матричным и матрично-матричным умножением чисел с плавающей запятой. Современные процессоры не оптимизированы для подобных вычислений. Гораздо лучше традиционных процессоров для этих задач подходят GPU — архитектуры, изначально созданные для работы с компьютерной графикой, а потому более подходящие для тензорных вычислений, однако и они не дают желаемого результата. Число параметров в современных формальных нейронных сетях может достигать сотен миллионов [2; 4].

Спайковые нейронные сети имеют ряд преимуществ перед формальными нейронными сетями.

- С помощью спайковых нейронных сетей возможно более богатое динамичное кодирование паттернов в непрерывном времени, которое, как правило, недоступно для формальных нейронных сетей [5].
- Для обучения спайковых нейронных сетей могут применять локальные алгоритмы [6—8], которые используют для обновления веса каждой связи лишь информацию от соединяемой данной связью нейронов. Напротив, при обучении формальных нейронных сетей для обновления веса каждой связи используется информация о всех нейронных связях, стоящих между этой связью и выходом сети. Таким образом, сами по себе алгоритмы обучения спайковых нейронных сетей являются более эффективными, чем алгоритмы обучения формальных нейронных сетей.
- Более того, эти алгоритмы могут использовать обучение без учителя (unsupervised learning), для которого не требуется ручной

разметки данных, как в случае использования методов обратного распространения ошибки.

- Спайковые нейронные сети могут быть реализованы на специализированном сверхэнергоэффективном (достигается за счет сокращения количества актов и средней длины передачи сигналов, а также пространственной и временной разреженности спайков в сети) нейроморфном процессоре. Такая реализация вместе с чисто алгоритмическим преимуществом спайковых нейронных сетей дает также существенный выигрыш в производительности. [9; 10].
- Спайковые нейронные сети в большей степени, чем формальные нейронные сети, биологически корректно моделируют взаимодействия нейронов, что может использоваться для биологических симуляций нервной системы в целях не только биоинформатических, но также биофизических и медицинских аспектов функционирования.

Представляет интерес изучение известных архитектур спайковых нейронных сетей и их топологий, в том числе классических архитектур формальных нейронных сетей — сверточной, локально соединенной и полносвязной [2]. Более того, введение дополнительных рекуррентных связей конкуренции в этих архитектурах способствует лучшему разделению признаков в процессе обучения [11; 12]. Полносвязная сеть — максимально простая модель, которая может использоваться в качестве сравнительного эталона. Сверточная и локально соединенная архитектуры используют свертку — операцию, позволяющую более эффективно (по сравнению с полносвязной сетью) выделять важные признаки. Локально соединенная архитектура особенно интересна тем, что легко реализуется аппаратно, так как, в отличие от сверточной сети, не использует общих синаптических весов. Локально соединенная архитектура имеет значительно меньшее количество параметров, чем полносвязная и в то же время позволяет обучать уникальный набор признаков для каждого рецептивного поля, в отличие от сверточных сетей. В то же время, сверточные сети

обладают трансляционной инвариантностью — полезным свойством для обработки изображений. Наличие конкуренции в нейронных сетях позволяет делать их признаки более независимыми. Поэтому для экспериментов были выбраны локально соединенные и сверточные сети со связями конкуренции, а в качестве базовой модели — полносвязная сеть со связями конкуренции.

На сегодняшний день разрабатывается несколько десятков нейроморфных процессоров для аппаратной реализации спайковых нейронных сетей. К одним из них относятся: TrueNorth, разработанный IBM (2014), созданная совместными усилиями Netflix и Google платформа SpiNNaker (2015), проект Intel — Loihi (2017) и Akida — нейроморфный процессор от BranChip (2020). В таблице 1 проведено сравнение данных аппаратных нейроморфных решений по ключевым параметрам, а также указаны аналогичные параметры человеческого мозга.

Таблица 1. Сравнительная таблица аппаратных реализаций спайковых нейронных сетей. Указано энергопотребление при решении типичной задачи машинного обучения.

Название	Нейроны	Синапсы	Энергопотребление, мВт
Мозг человека [13]	$\approx 16 \cdot 10^9$	$\approx 1 \cdot 10^{14}$	20000
TrueNorth [14]	$1 \cdot 10^6$	$256 \cdot 10^6$	$64^{(1)}$
SpiNNaker [15]	$1 \cdot 10^9$	$1 \cdot 10^{12}$	$1000^{(2)}$
Loihi [16]	$130 \cdot 10^3$	$130 \cdot 10^6$	неизвестно
Akida [17]	$1.2 \cdot 10^6$	$10 \cdot 10^9$	$\sim 10^{(3)}$

Все эти проекты разрабатываются под современные нужды и уже способны применяться в исследованиях или индустрии, что еще раз подчеркивает важность исследования спайковых нейронных сетей.

В настоящей работе:

1. Изучается влияние обучения связей конкуренции [11; 12; 18] между нейронами на точность распознавания образов в задаче классификации рукописных изображений цифр из датасета MNIST [19] при обучении без учителя для архитектуры локально соеди-

<sup>(1)</sup>Для обработки видео разрешением  $400 \times 240$  пикселей 30 кадров/с.

<sup>(2)</sup>При полной загрузке.

<sup>(3)</sup>Энергопотребление может быть от 1 мкВт до 10 мВт в зависимости от сложности задачи.

ненной сети (Locally Connected Spiking Neural Network, LCSNN) [20].

2. Проводится сравнение этой архитектуры со сверточной сетью (Convolution Spiking Neural Network, CSNN) и полносвязной сетью (Fully Connected Spiking Neural Network, FCSNN).

# 1 Обучение спайковой нейронной сети с необучаемой конкуренцией локальных рецептивных полей

## 1.1 Описание задачи

Для работы была выбрана классическая задача машинного обучения — задача классификации изображений рукописных цифр из набора данных MNIST. MNIST состоит из размеченных обучающей и тестовых выборок объемами 60000 и 10000 изображений. Изображения имеют размер  $28 \times 28$  пикселя и являются черно-белыми. Из-за необходимости калибровки сетей (1.4.1) обучающая выборка была разбита на 50000 изображений для обучения (обучающая выборка) и 10000 изображений для калибровки (калибровочная выборка).

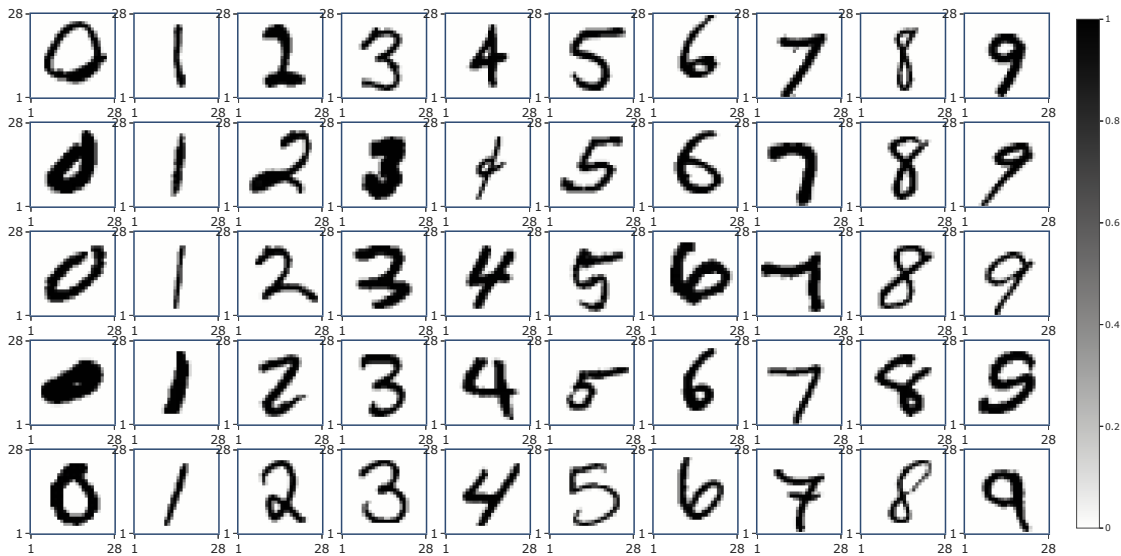


Рис. 1. Изображения MNIST

В этой работе изображения обрезаются так, что используется только центральная область  $20 \times 20$  пикселей.



## 1.2 Особенности архитектуры сетей с конкуренцией

В данной работе изучаются сети с использованием локально соединенных, полносвязных и сверточных слоев.

- **Полносвязный слой** - в нем каждый нейрон соединен с каждым нейроном из предыдущего слоя.
- **Сверточный слой** - в нем нейроны разделены на каналы. Веса всех нейронов в одном канале совпадают. Каналы равноправны, а каждый нейрон в канале соединен только с некоторой областью (прямоугольной в случае двумерного слоя) [21].
- **Локально соединенный слой** - он совпадает со сверточным слоем, но каждый нейрон имеет свои уникальные веса [20].

Топология сетей с конкуренцией включает в себя равноправные (расположенные в одном слое и имеющие аналогичные связи) нейроны, соединенные друг с другом синоптическими связями с отрицательным весом. Так как общий вид этих сетей совпадает, рассмотрим его на примере сети с локально соединенным слоем.

Нейросетевая архитектура LCSNN вдохновлена строением зрительной коры мозга.  $Y$  слой сети состоит из  $n$  каналов, каждый из которых является локально соединенным с  $X$  слоем нейронов.

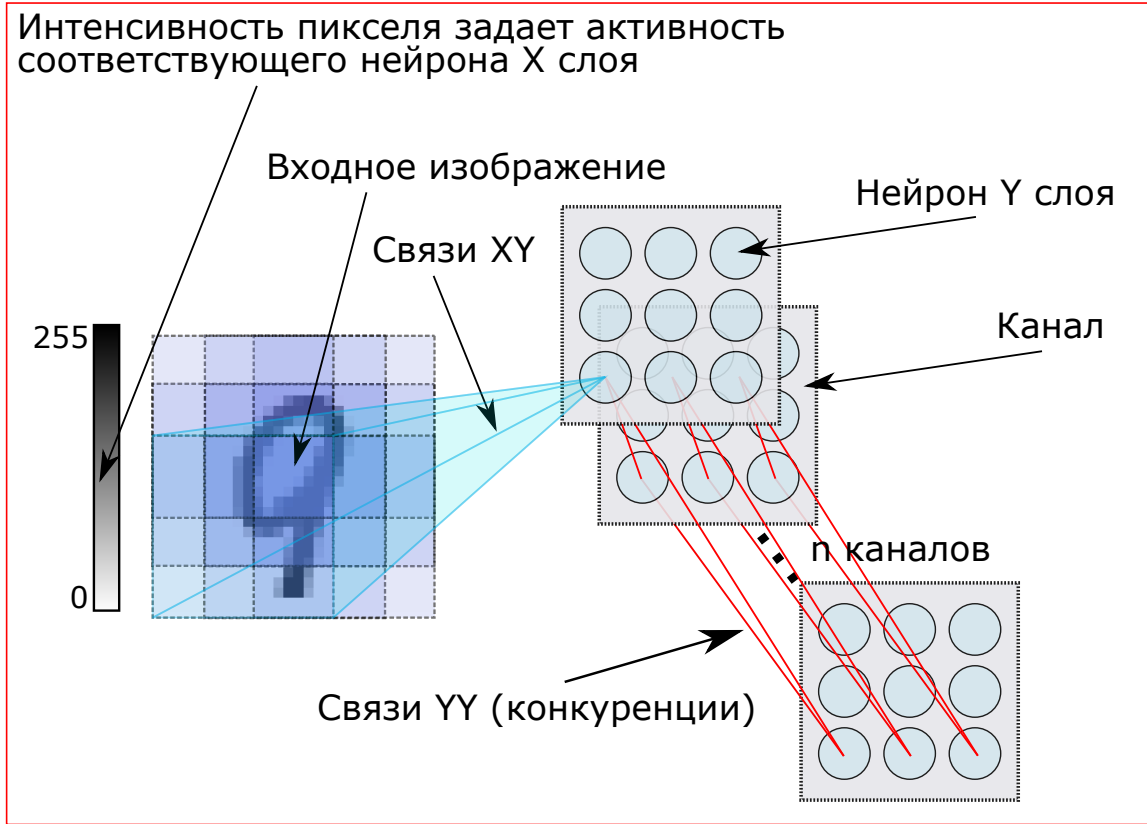


Рис. 2. Схема архитектуры LCSNN

Нейроны, имеющие общие рецептивные поля, соединяются  $YY$  связями конкуренции. Такие связи имеют отрицательные веса, а значит, негативно влияют на активность. Нейроны, не имеющие общего рецептивного поля (а значит, реагирующие на разные области изображения) не конкурируют между собой. Связи конкуренции вводятся для улучшения разделения нейронов по выучиваемым признакам. Аналогичная архитектура может быть построена для сетей с использованием сверточных или полносвязных слоев.

В этой работе используется Adaptive Integrate-And-Fire (ALIF) модель нейронов. Динамика потенциала нейрона в этой модели задается уравнением

$$\tau_v \frac{dv(t)}{dt} = -v(t) + v_{rest} + I(t) \cdot R, \quad (1)$$

где  $I(t)$  — ток, накопившийся в нейроне к моменту времени  $t$ ,  $v_{rest}$  — уровень релаксации,  $\tau_v$  — временная константа симуляции, а  $R$  — размерный коэффициент, численно равный 1.

Таким образом, потенциал нейрона сам по себе стремится вернуться в состояние релаксации, так как если  $I(t) = 0$  (нет входящих спайков) и  $v(t) = v_{rest}$ , то производная  $\frac{dv(t)}{dt} = 0$ , то есть потенциал нейрона будет оставаться постоянным на уровне  $v_{rest}$ .

Порог активации  $v_{thresh}$  у ALIF нейрона не является константой, а немного повышается при каждом спайке, релаксируя затем к своему начальному значению  $\theta_0$ . Динамика порога активации задается следующими уравнениями:

$$v_{thresh} = \theta_0 + \theta(t), \quad (2)$$

где  $\theta_0$  — начальный порог активации,  $\theta(t)$  — адаптивная добавка, которая вычисляется из условия

$$\tau_v \frac{d\theta(t)}{dt} = -\theta(t) \quad (3)$$

Также после испускания каждого спайка адаптивная добавка  $\theta(t)$  повышается на  $\theta_{plus}$ .

Таким образом, порог активации нейрона сам по себе стремится вернуться к начальному значению, так как при  $\theta(t) = \theta_0$ , то производная  $\frac{d\theta(t)}{dt} = 0$ , то есть порог активации нейрона будет оставаться постоянным на уровне  $\theta_0$ .

После испускания каждого спайка наступает короткий период рефрактерности, при котором на протяжении времени  $t_{refract}$  потенциал нейрона остается на уровне  $v_{reset}$ .

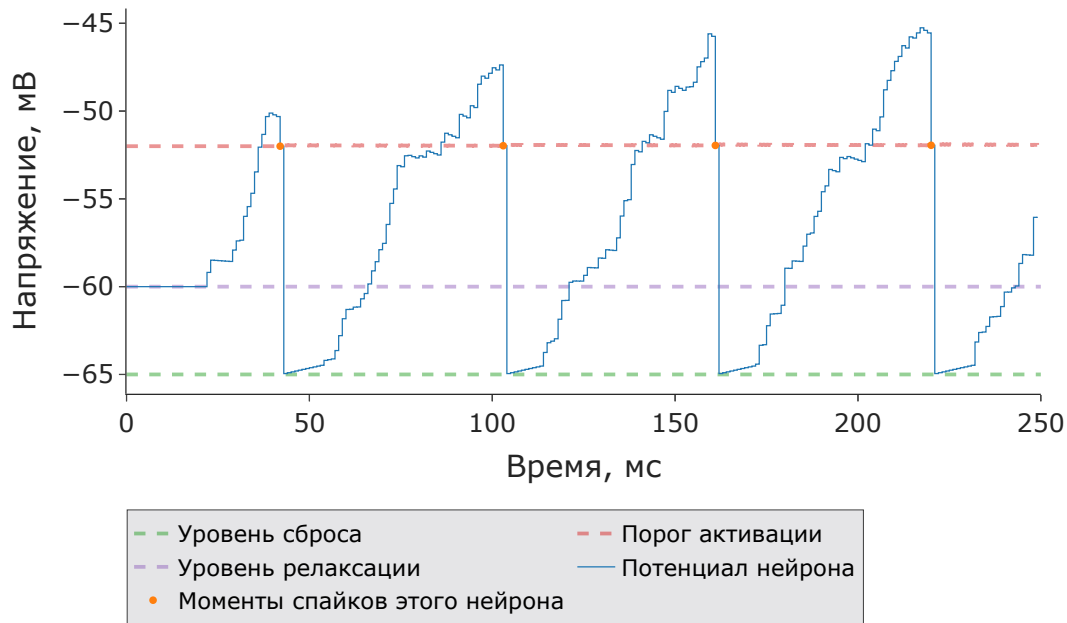


Рис. 3. Типичная зависимость потенциала порогового интегратора с утечкой и адаптивным порогом от времени. Красными точками отмечены спайки нейрона. Видны скачки порога активации после испускания спайков и его постепенное возвращение к исходному уровню. *Примечание: здесь исключительно для наглядности используется такое большое значение  $\theta_{plus}$ , в реальных экспериментах его влияние менее значительно.*

Начальные веса связей задаются случайными числами из равномерного распределения.

### 1.3 Обучение прямых связей

Для уменьшения числа параметров модели изображения обрезаются до размера  $20 \times 20$  пикселей. Края изображений часто практически

пусты, поэтому эта операция практически не влияет на объем информации, доступный сети. Для каждого изображения при помощи распределения Пуассона с математическим ожиданием, пропорциональным интенсивности соответствующего пикселя, генерируются  $X$  спайки. Обучение связей  $XY$  производится по правилу STDP [6]. Это биологически инспирированное правило обучения без учителя [6]. При получении пре-спайка и испускании пост-спайка вес  $w$  связи, по которой пришел пре-спайк, увеличивается на  $\Delta w$ , где

$$\Delta w = \begin{cases} A_+ \cdot e^{-\frac{t_{pre}-t_{post}}{\tau_+}}, & t_{pre} - t_{post} > 0 \\ A_- \cdot e^{-\frac{t_{pre}-t_{post}}{\tau_-}}, & t_{pre} - t_{post} < 0 \end{cases} \quad (4)$$

Заметим, что

$$\begin{cases} A_+ > 0 \\ A_- < 0 \end{cases}$$

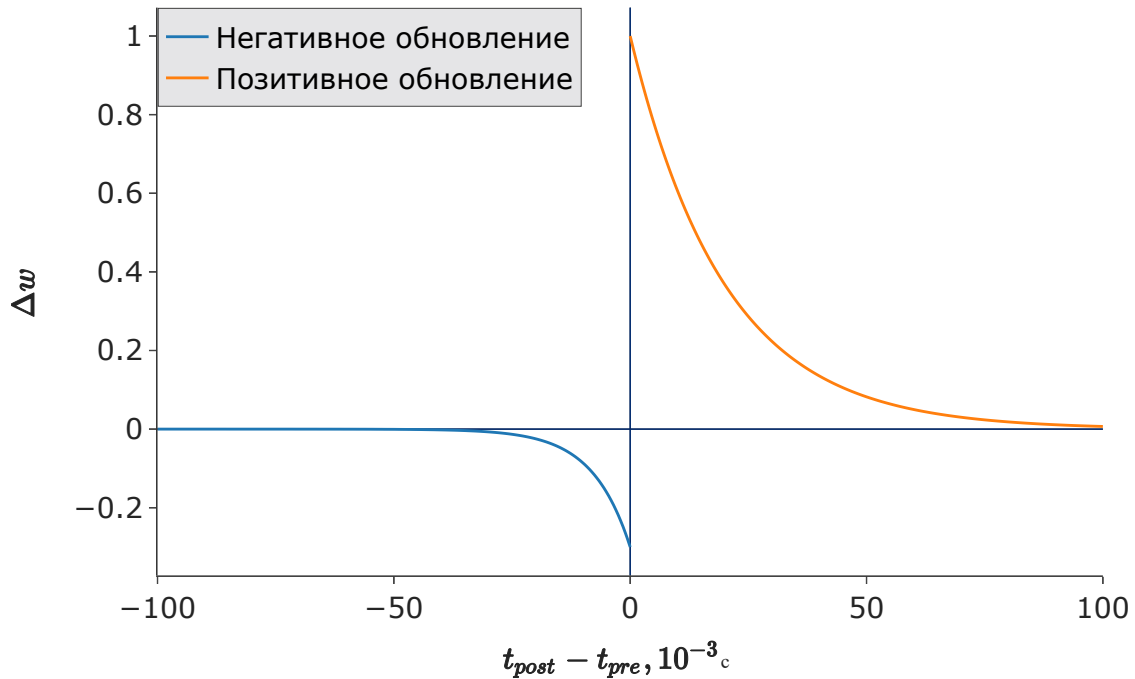


Рис. 4. Правило STDP. График зависимости изменения веса от разности времени регистрации пост- и пре-спайков.

Таким образом, в процессе обучения у каждого нейрона увеличивается вес связей, по которым систематически пре-спайк приходит перед излучением пост-спайка, и наоборот, вес уменьшается у тех связей, для которых такой закономерности не наблюдается. После такого обучения нейрон начинает активнее реагировать на пре-спайки от нейронов, соединенных с ним связями с большими весами, а значит, начинает сам испускать пост-спайк, если в короткий промежуток времени эти нейроны будут активны одновременно. Этим он объединяет их активность, посылая пост-спайк дальше по сети. Обратное правило (с отрицательными  $A_+$  и  $A_-$ ), называется правилом anti-STDP. В настоящей работе оно используется для обучения связей конкуренции.

После каждой итерации обучения производится нормализация весов — веса каждого нейрона умножаются на такое число, чтобы их

сумма стала равной определенной константе. Это делается для избежания возникновения слишком больших отдельных весов. Значение константы нормализации есть важный гиперпараметр модели, который подбирается для каждой конкретной архитектуры.

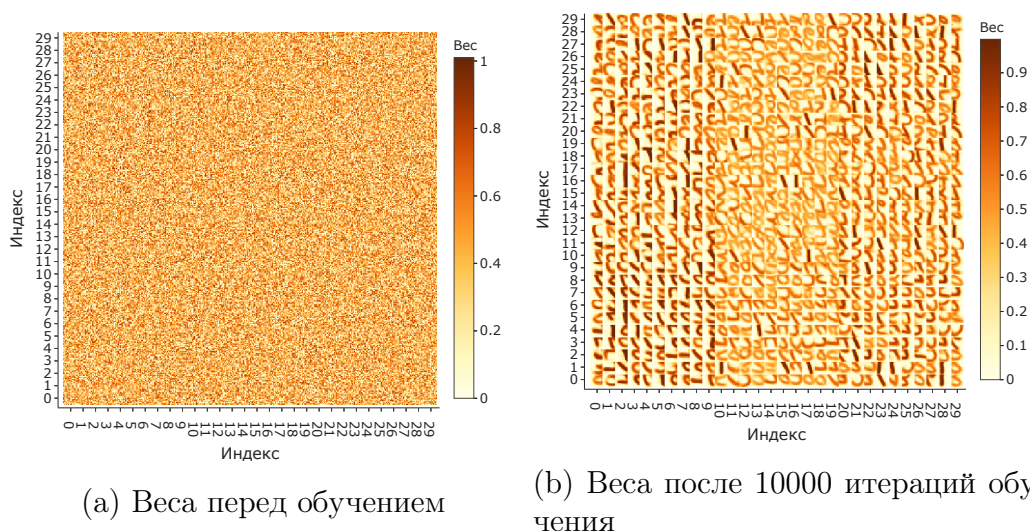


Рис. 5. Визуализация весов  $XU$  связей. Каждый квадратик  $12 \times 12$  соответствует весам одного  $U$  нейрона. Видно, что после обучения нейроны выучивают некоторые признаки, в которых явно угадываются элементы цифр.

## 1.4 Интерпретация активности спайковой нейронной сети

Для интерпретации активности нейронов  $U$  слоя сети использовалось несколько методов.

- Голосование патчей
- Общее голосование нейронов
- Голосование нейронов с предварительным отбором по спайкам
- Линейный классификатор

Преимущество первых трех методов заключается в их простоте. Однако, линейный классификатор значительно превосходит их по точности.

#### 1.4.1 Калибровка голосов

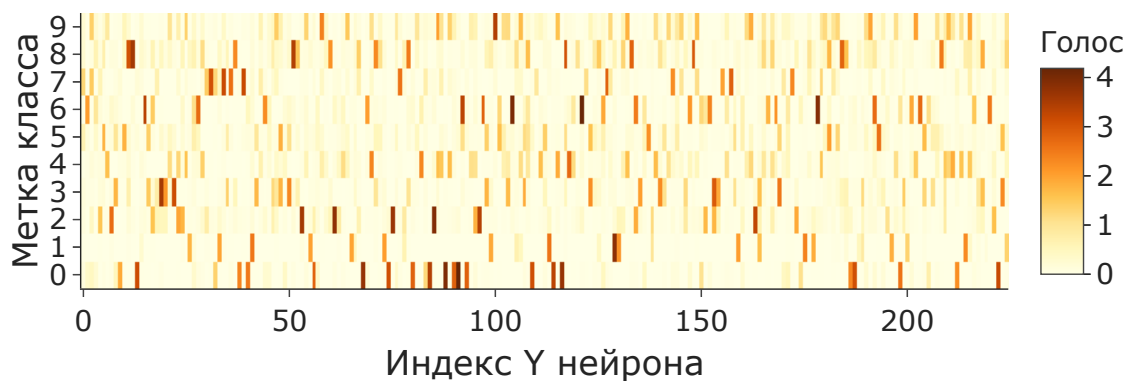
Для первых трех методов необходимо произвести калибровку голосов нейронов. Каждому нейрону  $Y$  слоя ставится в соответствие 10 чисел (голосов) для каждого возможного класса цифр (от 0 до 9). Голос вычисляется как усредненное по различным изображениям число спайков данного нейрона в ответ на демонстрацию сети данной цифры (на протяжении всего времени симуляции) —

$$vote = \frac{\sum_1^n \sum_0^{t_{max}} s_t}{n}, \text{ где}$$

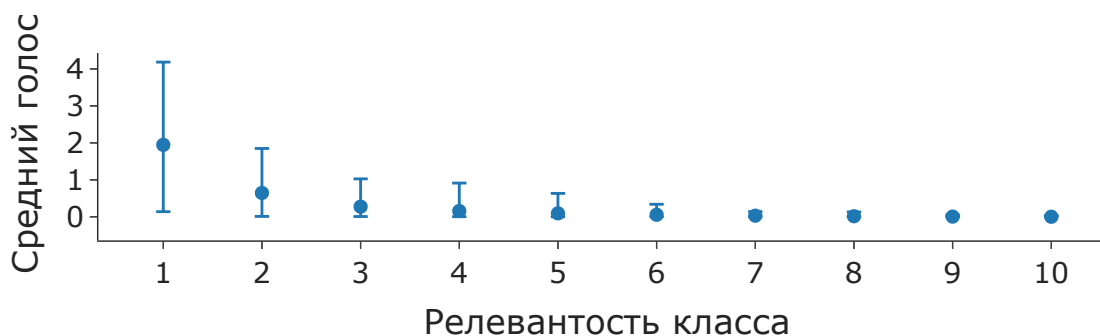
$s_t$  принимает значение 0 или 1 в зависимости от наличия спайка в данный момент времени,  $n$  — размер калибровочной выборки.

Для всех сетей использовалась калибровка на 10000 примерах из калибровочной выборки. Заметим, что калибровка не является частью обучения сети, так как она входит лишь в алгоритм интерпретации поведения сети. Эти голоса используются как мера уверенности нейрона в каждом из классов. После демонстрации изображения сети подсчитывается общее количество спайков для каждого нейрона  $Y$  слоя.





(а) Голоса нейронов.



(b) Ранжирование голосов нейронов по классам. Интервалами отмечены стандартные отклонения.

Рис. 6. Визуализация голосов нейронов  $Y$  слоя. Высокие значения соответствуют большой специализации нейрона на соответствующем классе.

Далее результатом будем называть произведение количества спайков нейрона на голос.

### Общее голосование

Ответом сети считается класс с максимальным результатом среди всех нейронов.

### Голосование патчей

Для каждого рецептивного поля ищется нейрон с максимальным результатом. Ответом сети считается класс с максимальным результатом среди таких нейронов для всех рецептивных полей.

### Отбор по спайкам

Для каждого рецептивного поля ищется нейрон с максимальным количеством спайков. Ответом сети считается класс с максимальным результатом среди таких нейронов для всех рецептивных полей.

### Линейный классификатор

Суммы спайков отдельных нейронов  $Y$  слоя используются для обучения с учителем линейного классификатора. Для обучения классификатора также используется калибровочная выборка.

### Оценка алгоритма интерпретации активности

Для оценки работы алгоритма интерпретации используется точность — отношение количества верно распознанных цифр к размеру тестовой выборки. В этой работе во всех случаях размер тестовой выборки составляет 10000.

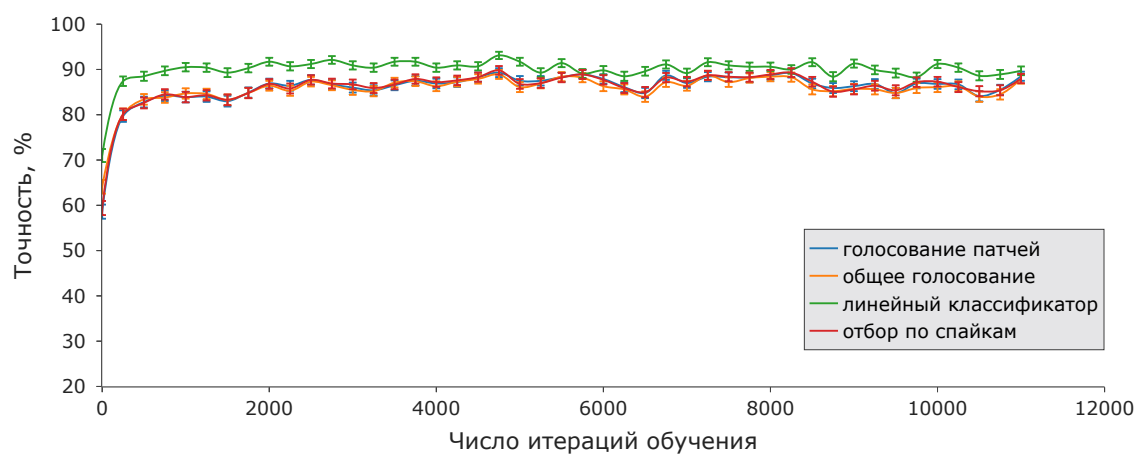
#### 1.4.2 Сравнение алгоритмов интерпретации активности сети

Были построены кривые обучения для различных алгоритмов интерпретации активности сети. Точность измерялась через каждые 250 итераций обучения. Для калибровки алгоритма интерпретации использовалась калибровочная выборка объемом 5000. Для измерения точности использовалась тестовая выборка объемом 1000.

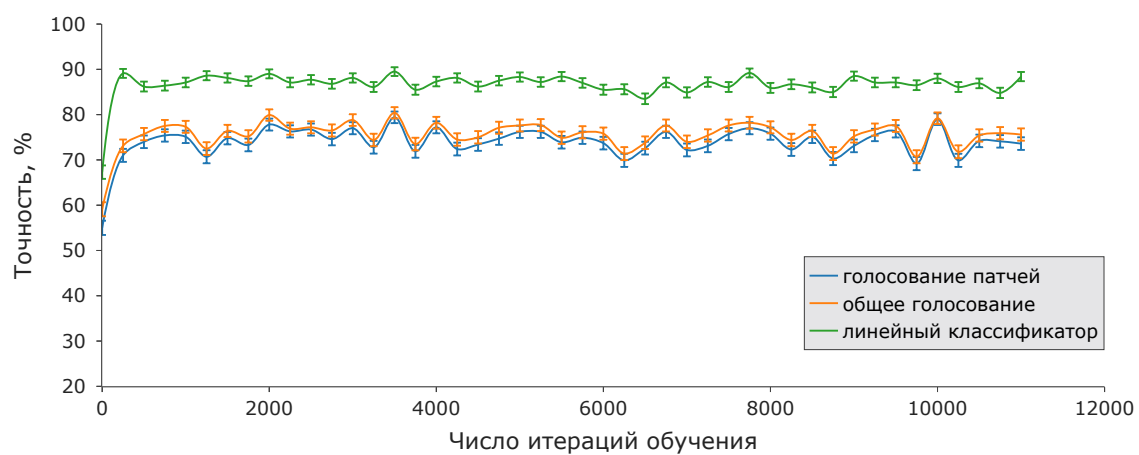
Видно, что через несколько тысяч итераций обучения точность распознавания выходит на плато насыщения, после чего уже не возрастает. Видно, что три метода голосования в целом не отличаются по точности, а линейный классификатор значительно превосходит их все. Заметим, что точность даже необученной сети может достигать

70% за счет того, что даже при случайно инициализации весов формируются отдельные нейроны, изначально более склонные к тому или иному классу. Из работы [20] известно, что при достаточно большом числе параметров локально соединенная сеть превосходит сверточную и полносвязную по скорости обучения, так как при каждой итерации обучения обновляется большее число параметров (активен один нейрон для каждого рецептивного поля). Здесь не наблюдается такого эффекта, так как представленные кривые обучения построены для недостаточно больших сетей (это вызвано вычислительной сложностью последнего). Интересно, что использование линейного классификатора не повышает точность распознавания для полносвязной сети — скорее всего, из-за простоты ее архитектуры.

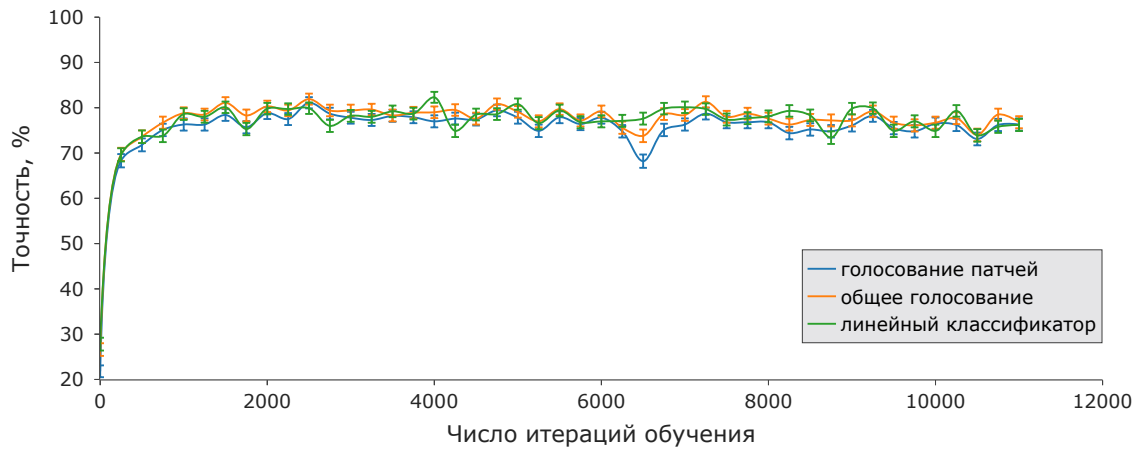
Линейный классификатор превосходит алгоритмы голосования по точности, так как является обобщением голосования в том смысле, что при его работе также используется сумма произведений весов (соответствуют голосам) на активности нейронов. Однако, при обучении линейного классификатора представляется возможным использование эффективных алгоритмов оптимизации (в том числе градиентных) его весов, тогда как при голосовании в качестве весов (голосов) используется некая эвристика - усредненная активность нейронов по классам.



(a) Кривая обучения LCSNN.



(b) Кривая обучения CSNN.



(с) Кривая обучения FCSNN.

Рис. 7. Сравнение кривых обучения различных архитектур спайковых нейронных сетей с конкуренцией. Приведены результаты для различных алгоритмов интерпретации активности. Отбор по спайкам проверялся только для LCSNN, потому что практически совпадает по результатам с голосованием патчей. В качестве погрешностей нанесены стандартные отклонения. Все эти сети имеют 100 каналов, размер свертки для LCSNN составляет 12, а для CSNN — 8.

## 1.5 Сравнение эффективности операции свертки и локального рецептивного поля

Были проведены эксперименты по измерению точности сетей с различными архитектурами. Для сверточных и полносвязных сетей вводились аналогичные связи конкуренции. Из-за очень высоких вычислительных нагрузок не ставилось задачи по нахождению параметров, идеально обеспечивающих максимальную точность для каждой архитектуры. Эти параметры были подобраны приблизительно, поэтому могли не попасть в максимум точности, однако величина расхождения не превышает 1-2%. Все точности посчитаны при помощи алгоритма интерпретации активности на основе голосования нейронов (учитывался лучший результат). Заметим, что при использовании линейного классификатора в качестве алгоритма интерпретации

удалось достигнуть 95% точности для локально соединенной сети из 1000 каналов с размером ядра 12.

Таблица 2. Результаты сравнения различных архитектур спайковых нейронных сетей. Для каждой сети точность измерялась  $N = 5$  раз. В таблице указаны среднее значение и стандартное отклонение для точности.

N	Конфигурация					Точность, %	
	Архитектура	Каналы	Ядро	Веса	Нейроны	Метод <sup>(1)</sup>	Метод <sup>(2)</sup>
1	LCSNN	1000	12	10287000	9000	$92.3 \pm 0.7$	$95.1 \pm 0.5$
2	LCSNN	100	12	218700	900	$87.5 \pm 0.9$	$91.5 \pm 0.6$
3	LCSNN	100	8	260800	1600	$82.9 \pm 0.6$	$88.1 \pm 0.7$
4	LCSNN <sup>(3)</sup>	25	12	37800	225	$82.3 \pm 1.0$	$88.2 \pm 0.6$
5	LCSNN	25	12	37800	225	$80.1 \pm 1.0$	$85.5 \pm 0.8$
6	LCSNN	25	8	35200	400	$73.6 \pm 1.0$	$80.3 \pm 0.7$
7	CSNN	169	12	255672	1521	$79.2 \pm 1.6$	$85.7 \pm 1.4$
8	CSNN	81	12	58464	729	$77.2 \pm 1.7$	$83.1 \pm 1.2$
9	CSNN	100	8	158464	1600	$77.4 \pm 1.9$	$82.1 \pm 1.3$
10	CSNN	25	12	5544	225	$65.8 \pm 0.7$	$77.1 \pm 0.6$
11	CSNN	25	8	9664	400	$63.1 \pm 1.2$	$75.8 \pm 0.5$
12	FCSNN	100	20	449100	100	$81.4 \pm 0.9$	$82.1 \pm 0.8$

Видно (№3 и №7 в таблице 2), что локально соединенная сеть превосходит сверточную сеть по точности даже при чуть превышающем числе параметров последней. Также видно, что не следует использовать слишком малый размер ядра свертки. Это приводит к выучиванию сетью менее значительных признаков. В классическом машинном обучении используются либо неглубокие сети с большими ядрами свертки, либо глубокие сети с маленькими ядрами свертки.

<sup>(1)</sup> Лучший алгоритм голосования

<sup>(2)</sup> Линейный классификатор

<sup>(3)</sup> Сеть с обучением связей конкуренции

Заметим, что спайковые нейронные сети могут достигать значительно больших точностей на MNIST. Для этого можно использовать:

- Сети с существенно большим числом весовых параметров, чем в данной работе (например, увеличив число каналов)
- Более глубокие сети с большим количеством слоев
- Другие алгоритмы обучения (например, обучение с учителем)

Таблица 3. Результаты других исследований спайковых нейронных сетей. Во всех используется датасет MNIST.

Статья	Архитектура	Обучение	Точность, %
[20]	Локальная + конкуренция	Без учителя	$95.07 \pm 0.63$
[22]	Полносвязная + конкуренция	Без учителя	95
[11]	Полносвязная + конкуренция	С учителем / с частичным привлечением учителя	95.4 / 72.1
[23]	Сверточная	С частичным привлечением учителя	$96.95 \pm 0.08$
[24]	Сверточная	С частичным привлечением учителя	$99.28 \pm 0.1$
[25]	Сверточная	С частичным привлечением учителя	$97.20 \pm 0.07$

Результат, полученный в настоящей работе, практически соответствует результату из [20]. Заметим, что при помощи признаков, обученных без учителя достигаются результаты, лишь немногим уступающие результатам, полученным при помощи обучения с учителем.

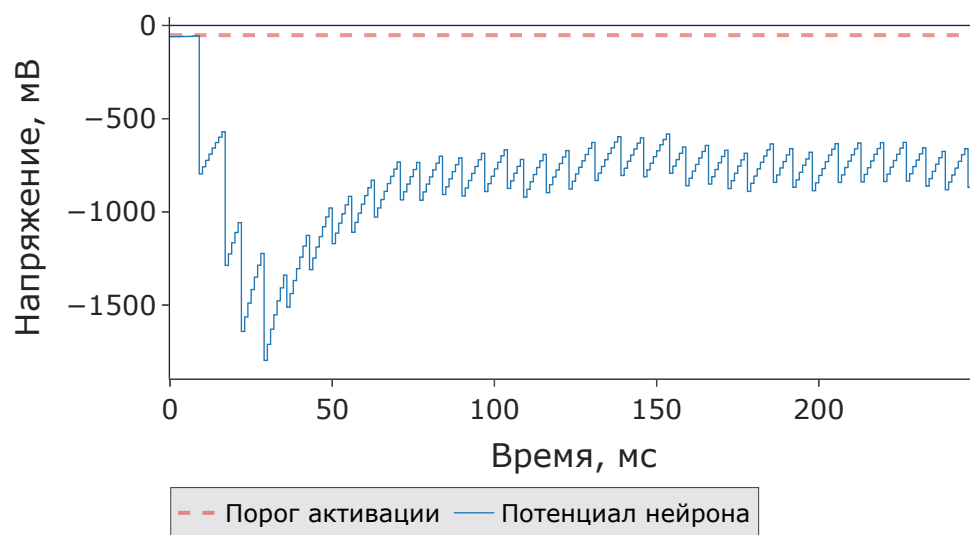
При этом лучшая локально соединенная сеть с конкуренцией локальных рецептивных полей из этой работы имеет  $10^7$  связей (из них связей прямого распространения —  $1 \cdot 10^6$ , связей конкуренции —  $9 \cdot 10^6$ ), по сравнению с  $4.6 \cdot 10^7$  (из них связей прямого распространения —  $0.5 \cdot 10^7$ , связей конкуренции —  $4.1 \cdot 10^7$ ) у полносвязной сети с конкуренцией из [22]. К тому же, обучение сети из [22] велось в течение  $1 \cdot 10^6$  итераций, тогда как в этой работе число итераций обучения не превосходит 5000.

## 2 Обучение связей конкуренции

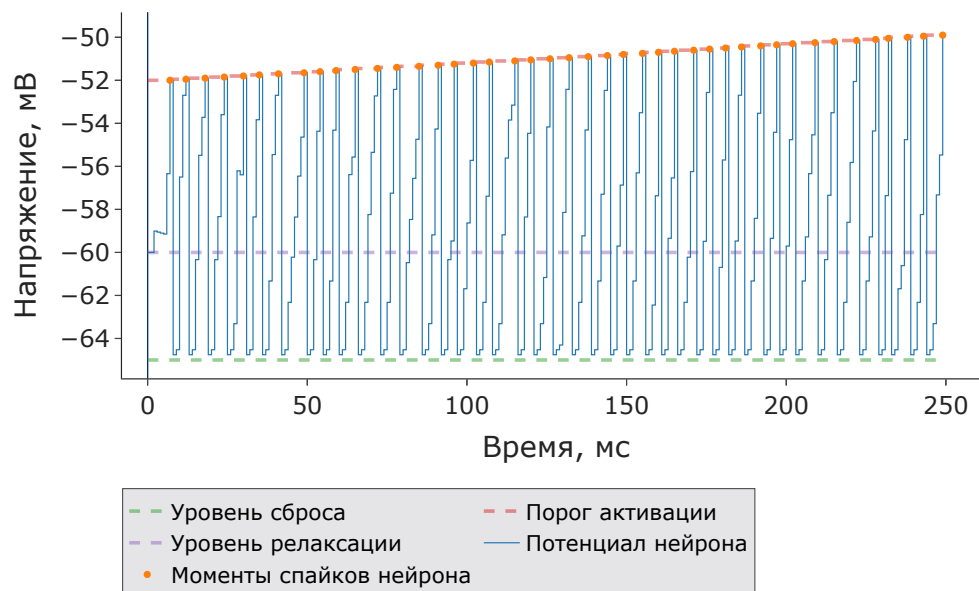
Связи  $YU$  существенно влияют на обучение связей  $XU$ . Большие по модулю значения способствуют вариативности в обучении  $Y$  нейронов, так для каждого рецептивного поля одновременно активными не могут быть нейроны, имеющие схожие веса  $XU$ . Малые по модулю значения не позволяют нейронам специализироваться.

Примечательно, что с такими большими по модулю значениями весов конкуренции для каждого рецептивного поля остается активным лишь один нейрон (или реже несколько нейронов), веса которого лучше всего соответствуют области представленного изображения, который подавляет активность остальных.



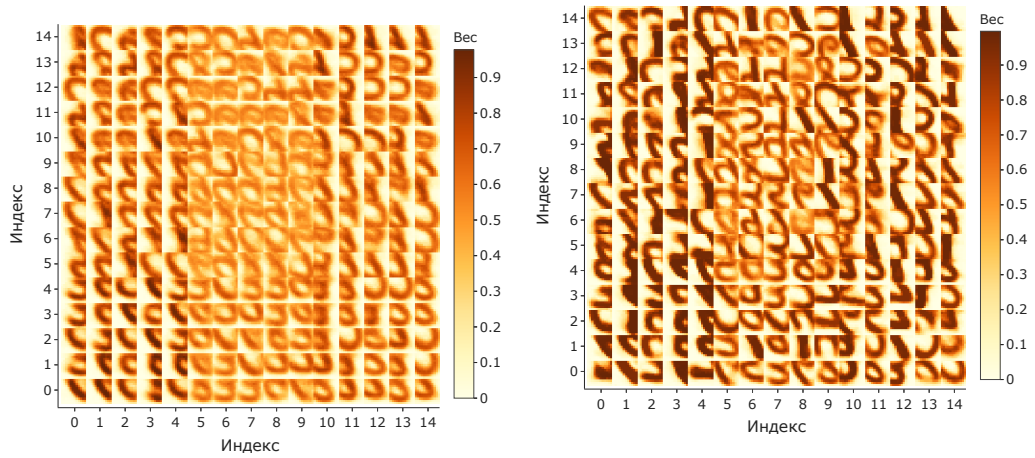


(а) Активность нейрона, который проиграл в конкуренции, подавлена другими нейронами.



(б) Активность нейрона, который выиграл в конкуренции. Присутствуют множество спайков.

Рис. 8. Влияние конкуренции на активность нейронов



(а) Слабо специализированные веса, вес конкуренции равен  $-10$ .  
 (б) Высоко специализированные веса, вес конкуренции равен  $-100$ .

Рис. 9. Влияние конкуренции на обучение связей  $XU$ .

Все сети, о которых шла речь до этого в работе, имели фиксированные веса конкуренции. Было обнаружено, что обучение весов конкуренции позволяет повысить точность сети. Для обучения связей  $YU$  используется правило anti-STP. При помощи варьирования параметров этого правила были получены различные распределения весов конкуренции.

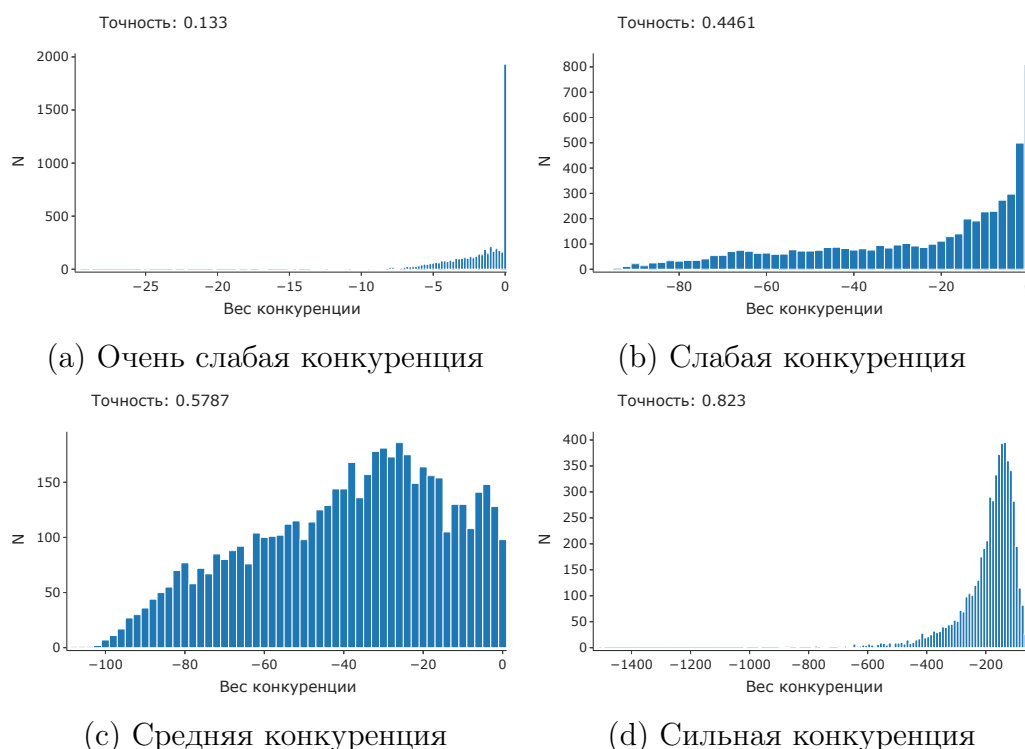
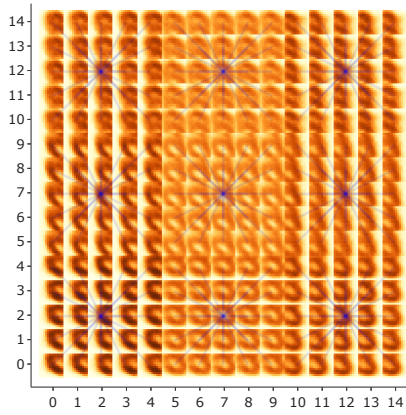


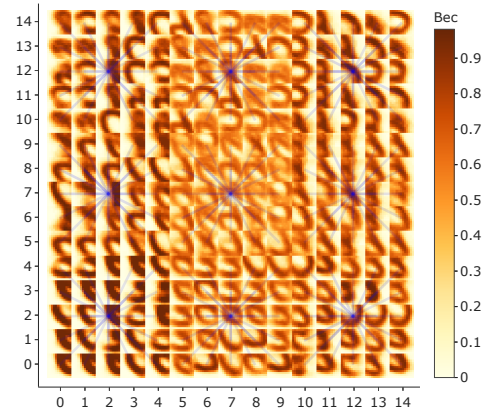
Рис. 10. Различные распределения весов конкуренции

Видно, что точность сети повышается при тяготении распределения весов конкуренции в сторону больших по модулю значений. Заметим, что целью являлось не нахождение параметров сети, обеспечивающих максимальную точность, а исследования влияния способности обучения конкуренции на точность сети с заданной конфигурацией остальных параметров.

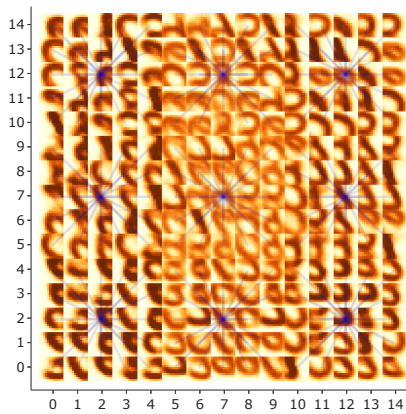
Примечательно, что не все связи  $YU$  получают большие по модулю значения. Это объясняется тем, что нейроны, специализирующиеся на значительно разных признаках, не нуждаются в конкуренции, так как они не проявляют высокую активность одновременно.



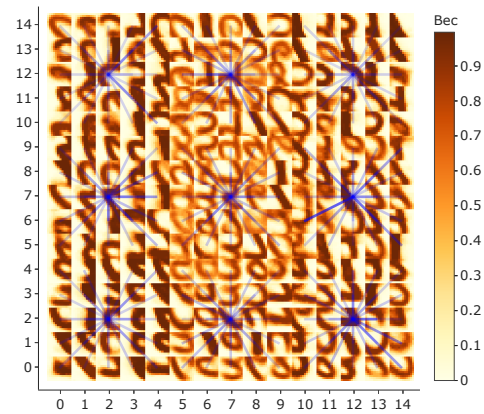
(a) Очень слабая конкуренция



(b) Слабая конкуренция



(c) Средняя конкуренция



(d) Сильная конкуренция

Рис. 11. Визуализация весов конкуренции поверх весов  $XU$ . Изображения соответствуют весам сетей с рисунка 10. Изображены только веса конкуренции для одного нейрона в каждом рецептивном поле для избегания загромождения визуализации. Насыщенный синий цвет соответствует большей по модулю конкуренции (используется среднее арифметическое между весами  $W_{ij}$  и  $W_{ji}$ ). Видно, что похожие признаки сильнее конкурируют между собой, чем различные.

В ходе оптимизации моделей была измерена точность большого количества сетей со различными конфигурациями гиперпараметров. Оказалось, что не существует четкого интервала ни для среднего ве-

са нормировки (определяющего средний  $XU$  связей для каждого  $U$  нейрона), для начального веса конкуренции, ни для параметров anti-STDP для обучения  $YU$  связей, который обеспечивал бы высокую точность классификации в не зависимости от значений других гиперпараметров. Это означает, что для максимизации точности не обязательно выбирать определенные значения этих гиперпараметров. Напротив, некоторые из них можно задать практически произвольно, а остальные подобрать. Это ценное наблюдение для реализации связей конкуренции на аппаратной основе.

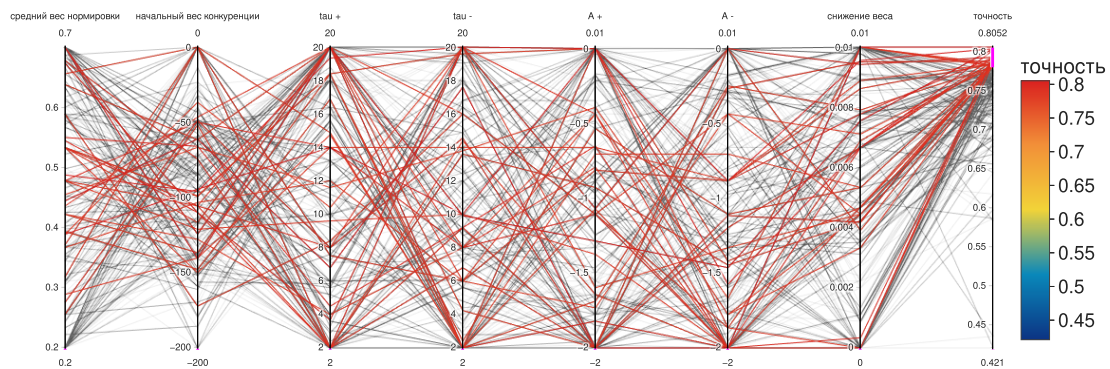


Рис. 12. Визуализация проверенного пространства гиперпараметров. 1,2 и 7 отвечают за значения весов, 3-6 - параметры anti-STDP. На последней оси указана точность сети с конфигурацией, соответствующей ломаной, проходящей через свои значения гиперпараметров. Видно, что ни на одной из осей нет области сосредоточения ломаных с высокой точностью (выделены красным).

## 2.1 Анализ обученных весов конкуренции

Были проведены эксперименты по ограничению значений весов конкуренции сверху и снизу. Оказалось, что как небольшие по модулю, так и большие по модулю веса конкуренции играют важную роль

в работе сети, так как ограничение как сверху, так и снизу отрицательно влияет на точность распознавания.

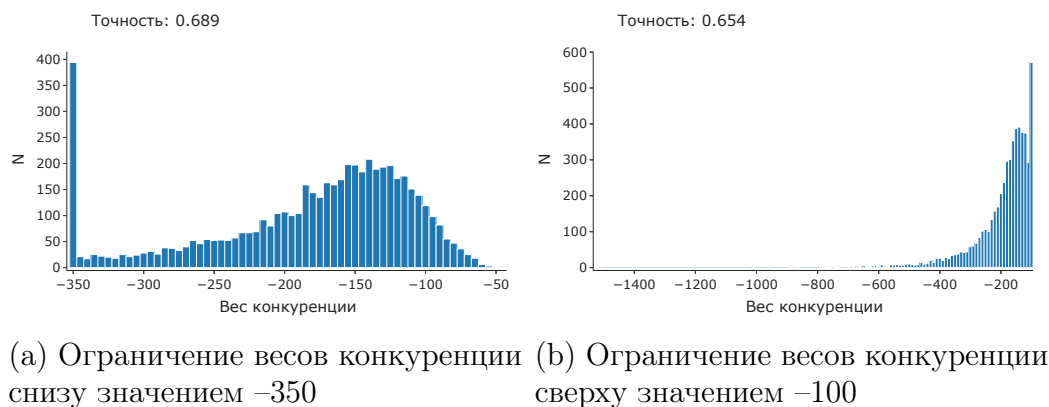


Рис. 13. Влияние ограничения значений весов конкуренции на точность. До ограничения точность этой сети составляла 0.823, а распределение ее весов конкуренции представлено на Рис. 10d.

Обучение конкуренции позволило достичь точности, на 2% превышающей точность сети такой же конфигурации, но с фиксированной конкуренцией значением  $-50$ . Обучение конкуренции производилось только на сетях из 25 каналов, так как его моделирование требует больших вычислительных ресурсов.

## Выводы

Было проведено сравнение локально соединенной спайковой нейронной сети со сверточной спайковой нейронной сетью. Оказалось, что локально соединенная архитектура значительно превосходит сверточную по точности при примерно одинаковом числе параметров.

Локально соединенная сеть – перспективная спайковая нейросетевая архитектура, подходящая для реализации на специализированном вычислительном нейрочипе. Способность быстро выходить на плато кривой обучения позволяет использовать локально соединенные сети для обучения на выборках сравнительно небольшого

объема (3000-5000). Использование дополнительного алгоритма интерпретации активности, такого как линейный классификатор, позволяет повысить эффективность сети. Заметим, что при этом сама сеть обучается без учителя. Возможно, схожие результаты можно получить и не используя обучение с учителем для интерпретации активности сети.

Число параметров (весов) сети определяет энергоэффективность мемристорного нейрочипа, так как каждый вес обычно представлен физическим элементом. Таким образом, это главный фактор, определяющий эффективность любой модели. Было показано, что локально соединенная сеть превосходит сверточную в точности при примерно одинаковом числе параметров.

Также было показано, что при помощи обучения связей конкуренции можно добиться большей точности, чем при фиксированных весах конкуренции.

В качестве перспектив для дальнейшего исследования можно отметить более глубокое исследование обучения конкуренции, а именно анализ распределений весов конкуренции, а также подбор оптимальных параметров обучения конкуренции. В этой работе не исследовалось обучение конкуренции у сетей с достаточно большим числом параметров и высокой точностью, так как такие задачи требуют слишком больших вычислительных ресурсов. Тем не менее, реализация такого обучения в аппаратном исполнении, например, на основе мемристорных кроссбаров [26], обеспечит возможность функционирования системы в реальном времени или быстрее, при этом с сохранением высоких показателей энергоэффективности и, как показывает настоящее исследование, с возможным улучшением точности распознавания SNN, потенциально способной достигнуть или даже превзойти текущие результаты the-state-of-the-art.



## Литература

1. Regularization of Neural Networks using DropConnect / L. Wan [и др.]. — 2013. — июнь. — URL: <http://proceedings.mlr.press/v28/wan13.html>.
2. A survey of the recent architectures of deep convolutional neural networks / A. Khan [и др.] // Artificial Intelligence Review. — 2020. — апр. — ISSN 1573-7462. — DOI: [10.1007/s10462-020-09825-6](https://doi.org/10.1007/s10462-020-09825-6). — URL: <http://dx.doi.org/10.1007/s10462-020-09825-6>.
3. *Edwards C.* Growing pains for deep learning // Commun. ACM. — 2015. — т. 58. — с. 14—16.
4. A State-of-the-Art Survey on Deep Learning Theory and Architectures / M. Z. Alom [и др.] // Electronics. — 2019. — март. — т. 8. — с. 292. — DOI: [10.3390/electronics8030292](https://doi.org/10.3390/electronics8030292).
5. Deep learning for time series classification: a review / H. Ismail Fawaz [и др.] // Data Mining and Knowledge Discovery. — 2019. — март. — т. 33, № 4. — с. 917—963. — ISSN 1573-756X. — DOI: [10.1007/s10618-019-00619-1](https://doi.org/10.1007/s10618-019-00619-1). — URL: <http://dx.doi.org/10.1007/s10618-019-00619-1>.
6. *Markram H., Gerstner W., Sjöström P. J.* A History of Spike-Timing-Dependent Plasticity // Frontiers in Synaptic Neuroscience. — 2011. — т. 3. — с. 4. — ISSN 1663-3563. — DOI: [10.3389/fnsyn.2011.00004](https://doi.org/10.3389/fnsyn.2011.00004). — URL: <https://www.frontiersin.org/article/10.3389/fnsyn.2011.00004>.
7. *Pehlevan C.* A Spiking Neural Network with Local Learning Rules Derived From Nonnegative Similarity Matching. — 2019. — arXiv: [1902.01429](https://arxiv.org/abs/1902.01429) [cs.NE].
8. *Baldi P., Sadowski P.* A theory of local learning, the learning channel, and the optimality of backpropagation // Neural Networks. — 2016. — нояб. — т. 83. — с. 51—74. — ISSN 0893-6080. — DOI: [10.1016/j.neunet.2016.07.006](https://doi.org/10.1016/j.neunet.2016.07.006). — URL: <http://dx.doi.org/10.1016/j.neunet.2016.07.006>.



9. Darwin: A neuromorphic hardware co-processor based on spiking neural networks / D. Ma [и др.] // Journal of Systems Architecture. — 2017. — т. 77. — с. 43–51. — ISSN 1383-7621. — DOI: [10.1016/j.sysarc.2017.01.003](https://doi.org/10.1016/j.sysarc.2017.01.003). — URL: <http://www.sciencedirect.com/science/article/pii/S1383762117300231>.
10. *Tang G., Shah A., Michmizos K. P.* Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM. — 2019. — arXiv: [1903.02504](https://arxiv.org/abs/1903.02504) [cs.R0].
11. *Demin V., Nekhaev D.* Recurrent Spiking Neural Network Learning Based on a Competitive Maximization of Neuronal Activity // Frontiers in Neuroinformatics. — 2018. — нояб. — т. 12. — с. 79. — DOI: [10.3389/fninf.2018.00079](https://doi.org/10.3389/fninf.2018.00079).
12. *Nekhaev D., Demin V.* Competitive Maximization of Neuronal Activity in Convolutional Recurrent Spiking Neural Networks. — 2020. — янв. — DOI: [10.1007/978-3-030-30425-6\\_30](https://doi.org/10.1007/978-3-030-30425-6_30).
13. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain / F. A. Azevedo [и др.] // Journal of Comparative Neurology. — 2009. — т. 513, № 5. — с. 532–541. — DOI: [10.1002/cne.21974](https://doi.org/10.1002/cne.21974). — eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cne.21974>. — URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cne.21974>.
14. A million spiking-neuron integrated circuit with a scalable communication network and interface / P. A. Merolla [и др.] // Science. — 2014. — т. 345, № 6197. — с. 668–673. — ISSN 0036-8075. — DOI: [10.1126/science.1254642](https://doi.org/10.1126/science.1254642). — eprint: <https://science.sciencemag.org/content/345/6197/668.full.pdf>. — URL: <https://science.sciencemag.org/content/345/6197/668>.
15. SpiNNaker: A 1-W 18-Core System-on-Chip for Massively-Parallel Neural Network Simulation / E. Painkras [и др.] // Solid-State Circuits, IEEE Journal of. — 2013. — авг. — т. 48. — с. 1943–1953. — DOI: [10.1109/JSSC.2013.2259038](https://doi.org/10.1109/JSSC.2013.2259038).

16. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning / M. Davies [и др.] // IEEE Micro. — 2018. — янв. — т. PP. — с. 1—1. — DOI: [10.1109/MM.2018.112130359](https://doi.org/10.1109/MM.2018.112130359).
17. Akida Neural Processor System-on-Chip. — 2020. — апр. — URL: <https://brainchipinc.com/akida-neuromorphic-system-on-chip/>.
18. Spiking Neural Networks Hardware Implementations and Challenges / M. Bouvier [и др.] // ACM Journal on Emerging Technologies in Computing Systems. — 2019. — июнь. — т. 15, № 2. — с. 1—35. — ISSN 1550-4840. — DOI: [10.1145/3304103](https://doi.org/10.1145/3304103). — URL: <http://dx.doi.org/10.1145/3304103>.
19. Gradient-based learning applied to document recognition / Y. Lecun [и др.] // Proceedings of the IEEE. — 1998. — т. 86, № 11. — с. 2278—2324.
20. Locally Connected Spiking Neural Networks for Unsupervised Feature Learning / D. J. Saunders [и др.]. — 2019. — arXiv: [1904.06269](https://arxiv.org/abs/1904.06269) [cs.NE].
22. *Diehl P., Cook M.* Unsupervised learning of digit recognition using spike-timing-dependent plasticity // Frontiers in Computational Neuroscience. — 2015. — т. 9. — с. 99. — ISSN 1662-5188. — DOI: [10.3389/fncom.2015.00099](https://doi.org/10.3389/fncom.2015.00099). — URL: <https://www.frontiersin.org/article/10.3389/fncom.2015.00099>.
23. *Tavanaei A., Maida A. S.* Multi-layer unsupervised learning in a spiking convolutional neural network. — 2017.
24. Training Deep Spiking Convolutional Neural Networks With STDP-Based Unsupervised Pre-training Followed by Supervised Fine-Tuning / C. Lee [и др.] // Frontiers in Neuroscience. — 2018. — т. 12. — с. 435. — ISSN 1662-453X. — DOI: [10.3389/fnins.2018.00435](https://doi.org/10.3389/fnins.2018.00435). — URL: <https://www.frontiersin.org/article/10.3389/fnins.2018.00435>.

25. *Tavanaei A., Kirby Z., Maida A. S.* Training Spiking ConvNets by STDP and Gradient Descent // 2018 International Joint Conference on Neural Networks (IJCNN). — 2018. — с. 1—8.
26. Review of memristor devices in neuromorphic computing: materials sciences and device challenges / Y. Li [и др.] // Journal of Physics D: Applied Physics. — 2018. — сент. — т. 51, № 50. — с. 503002. — DOI: [10.1088/1361-6463/aade3f](https://doi.org/10.1088/1361-6463/aade3f). — URL: <https://doi.org/10.1088/1361-6463/aade3f>.

Все материалы этой работы находятся в репозитории  
<https://github.com/danielgafni/bachelor>