

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени М.В.ЛОМОНОСОВА»

ФИЗИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА ОБЩЕЙ ФИЗИКИ И МОЛЕКУЛЯРНОЙ ЭЛЕКТРОНИКИ

БАКАЛАВРСКАЯ РАБОТА

**«МОДЕЛИРОВАНИЕ РАСПОЗНАВАНИЯ  
ОБРАЗОВ НА ОСНОВЕ ИМПУЛЬСНЫХ  
НЕЙРОННЫХ СЕТЕЙ С КОНКУРЕНЦИЕЙ  
ЛОКАЛЬНЫХ РЕЦЕПТИВНЫХ ПОЛЕЙ»**

Выполнил студент 406 группы  
Гафни Д.

---

подпись студента

Научный руководитель:  
к.ф.-м.н., н.с. Королёва А.В.

---

подпись научного руководителя

Научный консультант:  
Директор-координатор по направлению «природоподобные технологии»  
НИЦ «Курчатовский институт», к.ф.-м.н. Дёмин В.А.

---

подпись научного консультанта

Допущена к защите «\_\_»\_\_\_\_\_ 2020 г.

Заведующий кафедрой общей физики и молекулярной электроники  
д.ф.-м.н., профессор \_\_\_\_\_ П.К. Кашкаров  
подпись зав. кафедрой

Москва  
2020

# Содержание

<b>ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ И ОБОЗНАЧЕНИЯ</b>	<b>4</b>
<b>ВВЕДЕНИЕ</b>	<b>5</b>
Мотивация . . . . .	5
Постановка задачи . . . . .	6
<b>1 Спайковые нейронные сети</b>	<b>7</b>
1.1 Модели нейронов спайковой нейронной сети . . . . .	7
1.1.1 Пороговый нейрон . . . . .	7
1.1.2 Пороговый интегратор . . . . .	8
1.1.3 Пороговый интегратор с утечкой . . . . .	9
1.1.4 Пороговый интегратор с утечкой и адаптивным порогом	10
1.2 Архитектуры спайковых нейронных сетей . . . . .	12
1.2.1 Полносвязная сеть . . . . .	12
1.2.2 Сверточная сеть . . . . .	13
1.2.3 Локально соединенная сеть . . . . .	14
1.3 Алгоритмы обучения спайковых нейронных сетей . . . . .	14
1.3.1 STDP . . . . .	15
1.3.2 anti-STDP . . . . .	16
1.4 Современные аппаратные реализации спайковых нейронных сетей . . . . .	17
Выводы к разделу 2 . . . . .	17
<b>2 МОДЕЛИРОВАНИЕ ОБУЧЕНИЯ СПАЙКОВОЙ НЕЙРОН- НОЙ СЕТИ С КОНКУРЕНЦИЕЙ ЛОКАЛЬНЫХ РЕЦЕП- ТИВНЫХ ПОЛЕЙ</b>	<b>19</b>
2.1 Описание задачи классификации . . . . .	19
2.2 Особенности архитектуры . . . . .	19
2.3 Обучение связей $XU$ . . . . .	20
2.4 Интерпретация активности спайковой нейронной сети . . . . .	21
2.4.1 Калибровка голосов . . . . .	21
2.4.2 Сравнение алгоритмов интерпретации активности сети	23
2.5 Сравнение эффективности операции свертки и локального рецептивного поля . . . . .	24
2.6 Обучение связей $YU$ (конкуренции) . . . . .	26
Выводы к разделу 3 . . . . .	31
<b>ВЫВОДЫ</b>	<b>32</b>
<b>БЛАГОДАРНОСТИ</b>	<b>33</b>



# ИСПОЛЬЗУЕМЫЕ ТЕРМИНЫ И ОБОЗНАЧЕНИЯ

- **Нейронная сеть** — модель группы нейронов, соединенных связями. В **формальной нейронной сети** — **ФНС** — взаимодействие нейронов происходит мгновенно; в **спайковой нейронной сети** — **СНС** — нейроны обмениваются сигналами, и активность такой сети моделируется во времени.
- **Спайки: пост-спайк и пре-спайк** — сигналы, которыми обмениваются нейроны. Спайки распространяются по связям сети. Для синапса спайк пре-синаптического нейрона называется **пре-спайком**, а спайк пост-синаптического нейрона называется **пост-спайком**.
- **Вес связи** — вещественное число, являющееся коэффициентом передачи информации вдоль этой связи.
- **STDP** — локальное правило обучения без учителя весов связей спайковой нейронной сети [1]. **anti-STDP** — модификация STDP со значениями амплитуд противоположного знака. Применяется для обучения весов связей конкуренции.
- **Связи конкуренции** — связи, вес которых имеет отрицательное значение. Получение пре-спайка по такой связи подавляет активность пост-синаптического нейрона.
- **Обучение с учителем / без учителя** — в первом случае при обучении модели используется информация об истинных значениях целевой переменной, которую предсказывает модель; во втором случае используется лишь информация о признаках объектов обучения.
- **Полносвязное** соединение — тип связей между двумя слоями нейронов, при котором каждый нейрон первого слоя соединен с каждым нейроном второго слоя [2].
- **Свертка** — тип связи между двумя слоями нейронов, при котором каждый нейрон второго слоя соединяется лишь с нейронами первого слоя, лежащими в определенной области. При свертке веса связей нейронов второго слоя имеют общие значения [2].
- **Локальное соединение** — тип связи между двумя слоями нейронов, при котором каждый нейрон второго слоя соединяется лишь с нейронами первого слоя, лежащими в определенной области. В отличие от свертки, в локальном соединении каждый нейрон второго слоя связан с нейронами первого слоя уникальным набором весов [2].

# ВВЕДЕНИЕ

Импульсные (*спайковые*) нейронные сети (СНС) 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) [3].

Стандартные методы обучения весов связей, применяющиеся в формальных нейронных сетях (метод обратного распространения ошибки) затруднительно применять к СНС из-за их дискретной и распределенной во времени природы. Хотя такие проекты существуют [4; 5], больший интерес представляют локальные алгоритмы обучения [1], являющиеся как более биологически корректными, так и более энергоэффективными. Таким образом, исследование алгоритмов обучения СНС представляется важной задачей.

## Мотивация

Современные формальные нейронные сети отлично справляются со многими задачами машинного обучения [2; 6]. 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 [7]. 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 процессоров для этих задач подходят GPU — архитектуры, изначально созданные для работы с компьютерной графикой, а потому лучше подходящие для тензорных вычислений, однако и они не дают желаемого результата. Число параметров в современных формальных нейронных сетях может достигать сотен миллионов [2; 8].

Спайковые нейронные сети имеют ряд преимуществ перед формальными нейронными сетями.

- С помощью спайковых нейронных сетей возможно временное кодирование паттернов, которое, как правило, недоступно для формальных нейронных сетей.

- Для обучения спайковых нейронных сетей могут применять локальные алгоритмы [1; 9; 10], которые используют для обновления веса каждой связи лишь информацию, источником которой являются нейроны, которые эта связь соединяет. Напротив, при обучении формальных нейронных сетей для обновления веса каждой связи используется информация о всех 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 используют обучение без учителя (unsupervised learning), для которого не требуется ручной разметки данных.
- Спайковые нейронные сети могут быть реализованы на специализированном сверхэффективном (энергоэффективность достигается за счет пространственной и временной разреженности спайков в сети) нейроморфном процессоре. Такая реализация вместе с чисто алгоритмическим преимуществом спайковых нейронных сетей дает еще больший выигрыш в производительности и энергоэффективности. [3; 11].
- Спайковые нейронные сети биологически корректно моделируют взаимодействия нейронов, что может использоваться для биологических симуляций нервной системы.

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 [12; 13].

## Постановка задачи

В данной работе:

1. Изучается влияние обучения связей конкуренции [12—14] между нейронами на точность распознавания образов в задаче классификации рукописных изображений цифр из датасета MNIST [15] при обучении без учителя для архитектуры локально соединенной сети (Locally Connected Spiking Neural Network, LCSNN) [16].
2. Проводится сравнение этой архитектуры со сверточной сетью (Convolution Spiking Neural Network, CSNN) и полносвязной сетью (Fully Connected Spiking Neural Network, FCSNN).

# 1 Спайковые нейронные сети

Спайковая нейронная сеть — модель нейронной сети, элементами которой являются отдельные *нейроны* и *связи* (соответствующие синапсам реальных нейронов) между ними. Каждый нейрон имеет свой виртуальный *потенциал*, а каждая связь имеет некоторый *вес*. Нейроны обмениваются дискретными электрическими сигналами (*спайками*), имеющими очень короткую ( $\approx 1$  мс) длительность. Входящий спайк носит название *пре-спайка*, а исходящий — *пост-спайка*. Влияние пре-спайков на потенциал нейрона определяется значением веса межнейронной связи. При накоплении потенциала, превышающего определенный *порог активации*, нейрон сам испускает пост-спайк, а после сбрасывает свое напряжение до значения *потенциала сброса*. При запуске сети активность некоторых входных нейронов задается определенным образом, после чего проводится симуляция на протяжении некоторого времени ( $\approx 250$  мс), много большего длительности спайка.

## 1.1 Модели нейронов спайковой нейронной сети

Реальный нейрон мозга является очень сложным объектом, поведение которого до сих пор не до конца изучено. Для спайковых нейронных сетей используются упрощенные модели нейронов [17]. Перечислим некоторые из них.

### 1.1.1 Пороговый нейрон

Это наиболее упрощенная модель нейрона. В каждый момент времени нейрон осуществляет следующие действия:

1. Суммирует входящие сигналы

$$v(t) = v_{reset} + R \cdot \sum_{i=1}^n w_i I_i, \text{ где} \quad (1)$$

$v(t)$  — потенциал нейрона,  $v_{reset}$  — уровень сброса,  $w_i$  — вес  $i$ -ой связи,  $I_i$  — «ток»  $i$ -ой связи, принимающий значение 1, если от нее есть спайк и 0, если его нет, а  $R$  — размерный коэффициент, численно равный 1.

2. Осуществляет сравнение  $v(t)$  с  $v_{thresh}$  — значением порога активации. Если потенциал превосходит порог активации, то нейрон испускает спайк и сбрасывает свой потенциал до  $v_{reset}$ . Если потенциал не превосходит порог активации, то нейрон просто сбрасывает свой потенциал до значения  $v_{reset}$ .

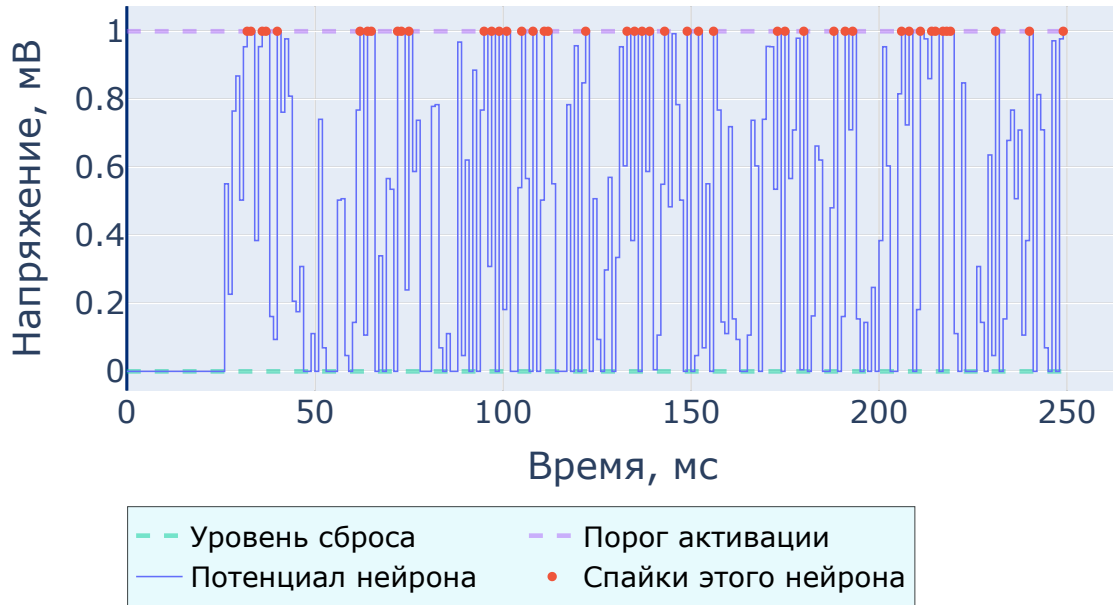


Рис. 1. Типичная зависимость потенциала порогового нейрона от времени. Красными точками отмечены спайки нейрона. В каждый момент времени потенциал нейрона равен взвешенной сумме входящих спайков.

### 1.1.2 Пороговый интегратор

Это немного усложненная модель нейрона (Integrate-and-Fire, IF), в которой нейрон приобретает способность накапливать входящие сигналы во времени. В каждый момент времени нейрон осуществляет следующие действия:

1. суммирует входящий сигнал

$$v(t) = v_{reset} + R \cdot \sum_{i=1}^n w_i I_i + R \cdot I(t), \text{ где} \quad (2)$$

$I(t)$  — накопленный к этому моменту ток от входящих спайков, а остальные переменные обозначены в выражении 1.

2. осуществляет сравнение  $v(t)$  с  $v_{thresh}$  — значением порога активации. Если потенциал превосходит порог активации, то нейрон испускает



спайк и сбрасывает свой потенциал до значения  $v_{reset}$ , а также сбрасывает  $I(t)$  до 0, после чего наступает короткий период рефрактерности, при котором на протяжении времени  $t_{refract}$  потенциал нейрона остается на уровне  $v_{reset}$ . Если же потенциал не превосходит порога активации, то нейрон просто обновляет значение  $I(t)$ .

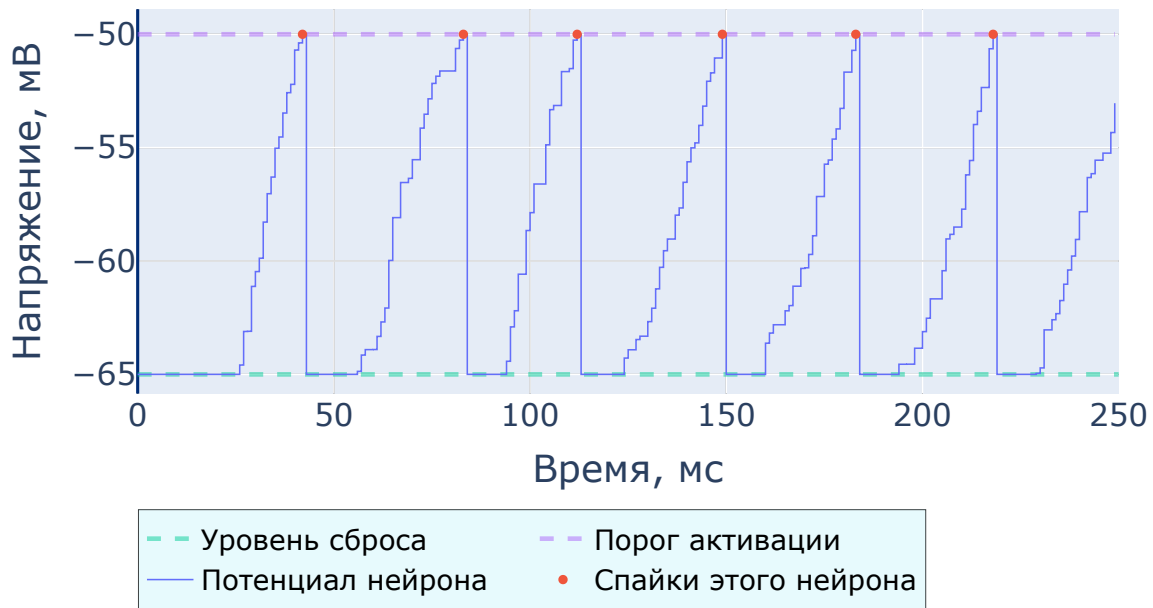


Рис. 2. Типичная зависимость потенциала порогового интегратора от времени. Красными точками отмечены спайки нейрона. В отличие от предыдущей модели, потенциал накапливается, пока не достигнет порога активации.

### 1.1.3 Пороговый интегратор с утечкой

Это еще более усложненная модель нейрона (Leaky-Integrate-and-Fire, LIF), в которой его потенциал приобретает «утечку», то есть стремится с течением времени сравняться с *уровнем релаксации*  $v_{rest}$ . В этой модели динамика потенциала задается следующим уравнением:

$$\tau_v \frac{dv(t)}{dt} = -v(t) + v_{rest} + I(t) \cdot R, \text{ где} \quad (3)$$

где  $\tau_v$  — временная константа симуляции, остальные переменные обозначены в выражениях 2 и 1.

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

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  $v_{rest}$ .



Рис. 3. Типичная зависимость потенциала порогового интегратора с утечкой от времени. Красными точками отмечены спайки нейрона. Видно, что потенциал нейрона сам по себе стремится к значению потенциала релаксации, чего не происходит в предыдущей модели.

#### 1.1.4 Пороговый интегратор с утечкой и адаптивным порогом

В данной работе используется именно эта модель (Adaptive Leaky Integrate-and-Fire, A-LIF), самая сложная из представленных. В этой модели динамика потенциала задается таким же уравнением, как и у LIF нейрона:

$$\tau_v \frac{dv(t)}{dt} = -v(t) + v_{rest} + I(t) \cdot R, \quad (4)$$

где  $I(t)$  — ток, накопившийся в нейроне к моменту времени  $t$ ,  $v_{rest}$  — уровень релаксации,  $\tau_v$  — временная константа симуляции.

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:

$$v_{thresh} = \theta_0 + \theta(t), \quad (5)$$

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

$$\tau_v \frac{d\theta(t)}{dt} = -\theta(t) \quad (6)$$

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  $v_{reset}$ .

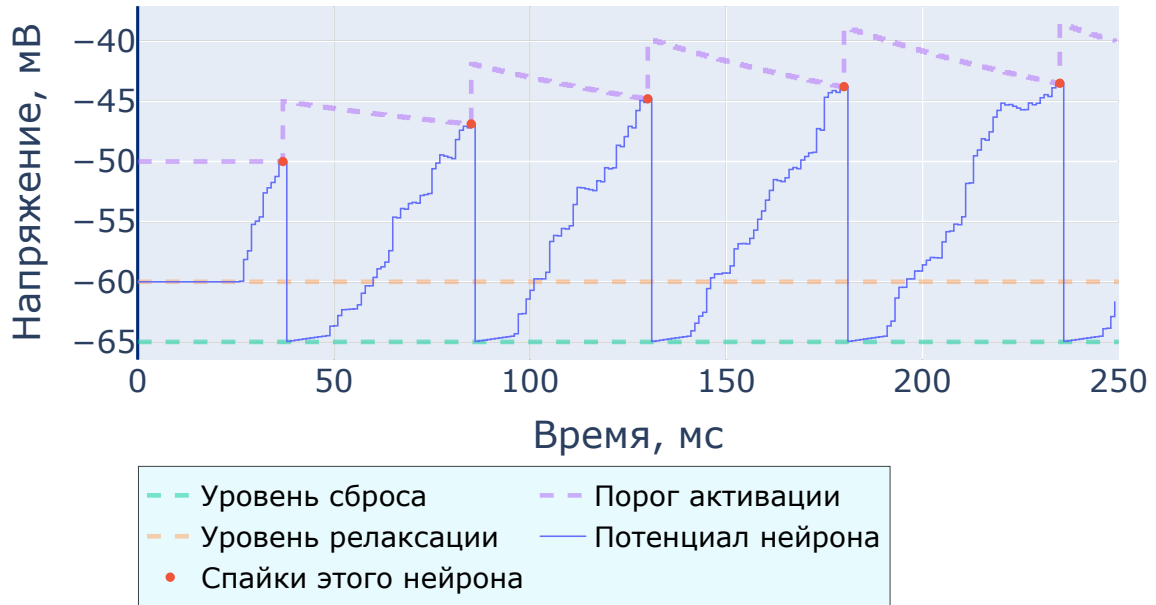


Рис. 4. Типичная зависимость потенциала порогового интегратора с утечкой и адаптивным порогом от времени. Красными точками отмечены спайки нейрона. Видны скачки порога активации после испускания спайков и его постепенное возвращение к исходному уровню.

*Примечание: здесь исключительно для наглядности используется такое большое значение  $\theta_{plus}$ , в реальных экспериментах его влияние менее значительно.*

## 1.2 Архитектуры спайковых нейронных сетей

Современные архитектуры формальных нейронных сетей [2] могут переноситься на спайковые нейронные сети. При этом для увеличения эффективности разделения нейронов по признакам используются связи конкуренции (Рис. 15).

### 1.2.1 Полносвязная сеть

В полносвязной сети два слоя  $X$  и  $Y$  имеют связи между всеми парами нейронов.

Число нейронов  $Y$  слоя и его структура задается произвольно. Число параметров (весов) такой сети равно  $\frac{XY}{2}$ .

### 1.2.2 Сверточная сеть

В сверточной сети для построение связей и формирования  $Y$  слоя применяется операция *свертки*. Пусть есть  $X$  слой нейронов  $N_1 \times N_2$ . Операция свертки определяется 4 параметрами: размером *ядра* свертки  $(k_1, k_2)$  и *шагом* свертки  $(s_1, s_2)$ .

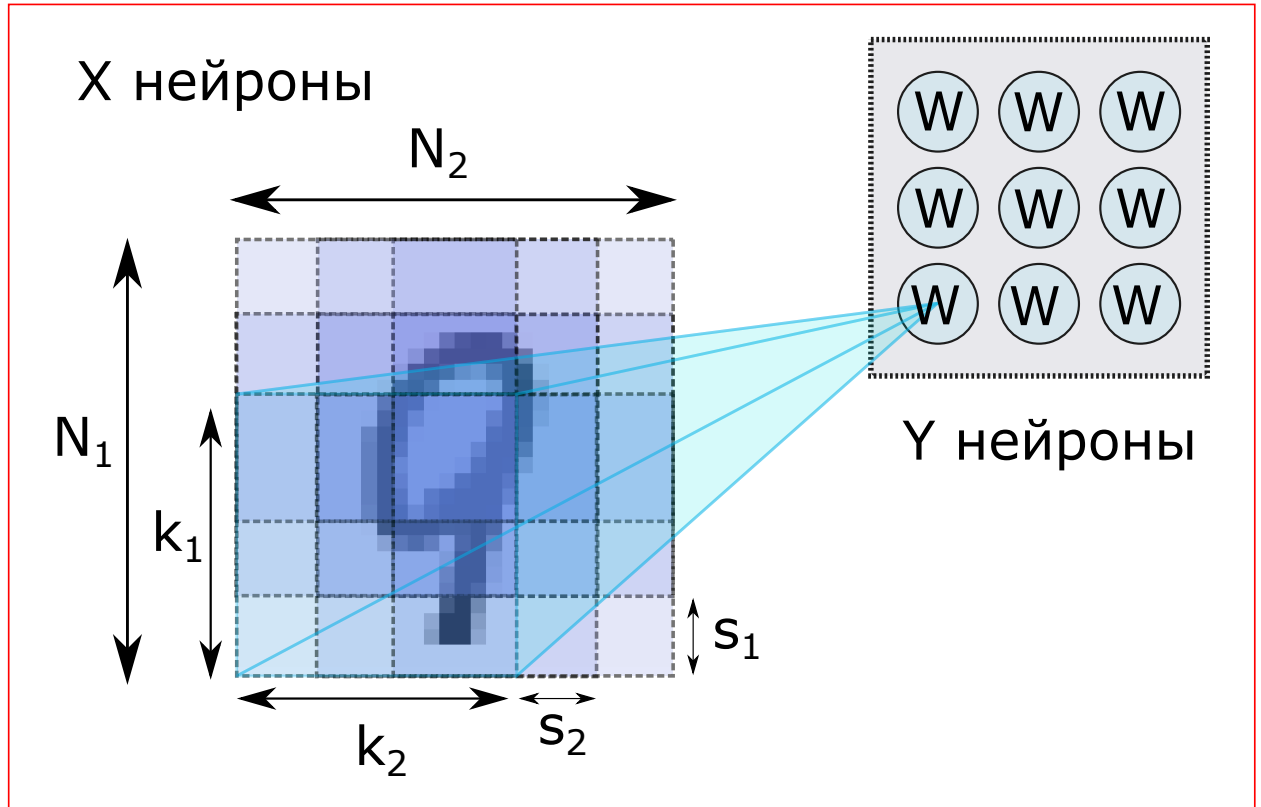


Рис. 5. Свертка

$Y$  нейроны также называются *сверточным* слоем. Все они имеют общие (численно совпадающие) веса  $W$  связей  $XY$  размерности  $(k_1 \times k_2)$ . Отличаются нейроны положением области  $X$  слоя, с которой имеются связи.

Размерность  $Y$  слоя равна  $(\frac{N_1 - k_1 - 2}{s_1} \times \frac{N_2 - k_2 - 2}{s_2})$ . Число параметров свертки равно  $k_1 \cdot k_2$ .

Заметим, что свертки могут применяться над предыдущим слоем любой размерности. Здесь была описана 2-D свертка, принцип ее работы аналогично переносится и на N-D.

В дальнейшем ядро свертки  $k_1 \times k_2$  будем также называть синонимами — «*область видимости*», «*рецептивное поле*». Также в этой работе для упрощения подбора оптимальных параметров используются только квадратные свертки, т.е.  $k_1 = k_2 \equiv k$ , а  $s_1 = s_2 \equiv s$ . Все сети в этой работе имеют шаг  $s = 2$ .

### 1.2.3 Локально соединенная сеть

Локально соединенная архитектура совпадает со сверточной с тем единственным отличием, что веса  $Y$  нейронов для нее не являются общими, то есть каждый  $Y$  нейрон имеет свои уникальные веса связей  $XU$ .

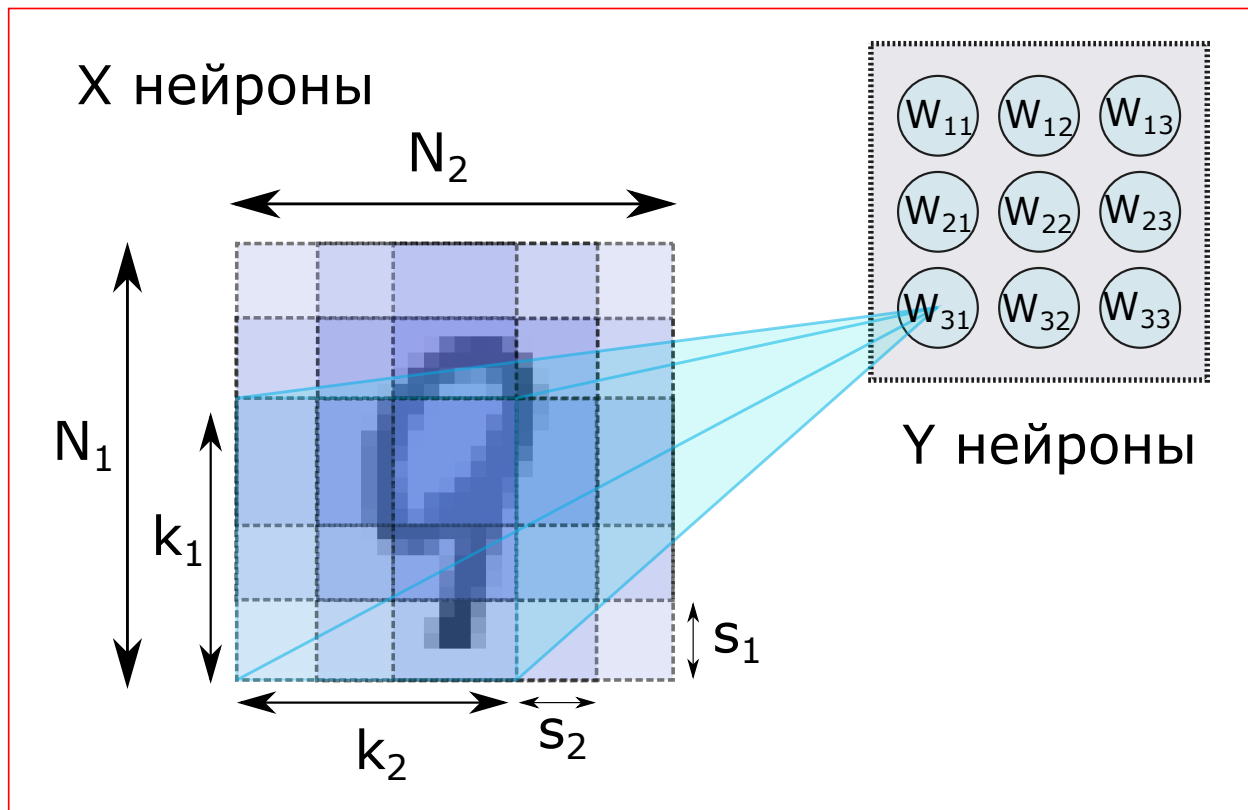


Рис. 6. Локальное соединение

Размерность  $Y$  слоя для локально соединенной сети равна  $(\frac{N_1 - k_1 - 2}{s_1} \times \frac{N_2 - k_2 - 2}{s_2})$ , число связей равно  $\frac{N_1 - k_1 - 2}{s_1} \cdot \frac{N_2 - k_2 - 2}{s_2} \cdot k_1 \cdot k_2$ .

Как и свертка, локально соединенный слой может быть аналогичным образом основан на слое любой размерности. Для локально соединенных сетей в этой работе используются только квадратные свертки (для упрощения подбора гиперпараметров), т.е.  $k_1 = k_2 \equiv k$ , а  $s_1 = s_2 \equiv s$ . Все сети в этой работе имеют шаг  $s = 2$ .

## 1.3 Алгоритмы обучения спайковых нейронных сетей

Процесс нахождения оптимальных значений весов связей сети называется обучением. Обучением может вестись как с учителем — если алгоритм обучения использует информацию об истинных значениях, предсказываемых сетью — так и без учителя. Преимущество первого класса алгоритмов обучения заключается в как правило лучших результатах, а недостаток — в необходимости предварительной разметки обучающей выборки

человеком, что не всегда представляется возможным. В свою очередь, обучения без учителя может давать немного худшие результаты в зависимости от задачи, однако не требует разметки данных, что часто позволяет подготавливать большие объемы обучающей выборки, чем при обучении без учителя. В данной работе для обучения используется правило STDP (Spike Timing Dependent Plasticity).

### 1.3.1 STDP

STDP — биологически инспирированное правило обучения без учителя [1]. При получении пре-спайка и испускании пост-спайка вес  $w$  связи, по которой пришел пре-спайк, увеличивается на  $\Delta w$ , где

$$\Delta w = \begin{cases} A_+ \cdot e^{-\frac{t_{pre}-t_{post}}{\tau_+}}, & t_{pre} - t_{post} > 0 \\ A_- \cdot e^{-\frac{t_{pre}-t_{post}}{\tau_-}}, & t_{pre} - t_{post} < 0 \end{cases} \quad (7)$$

Заметим, что

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

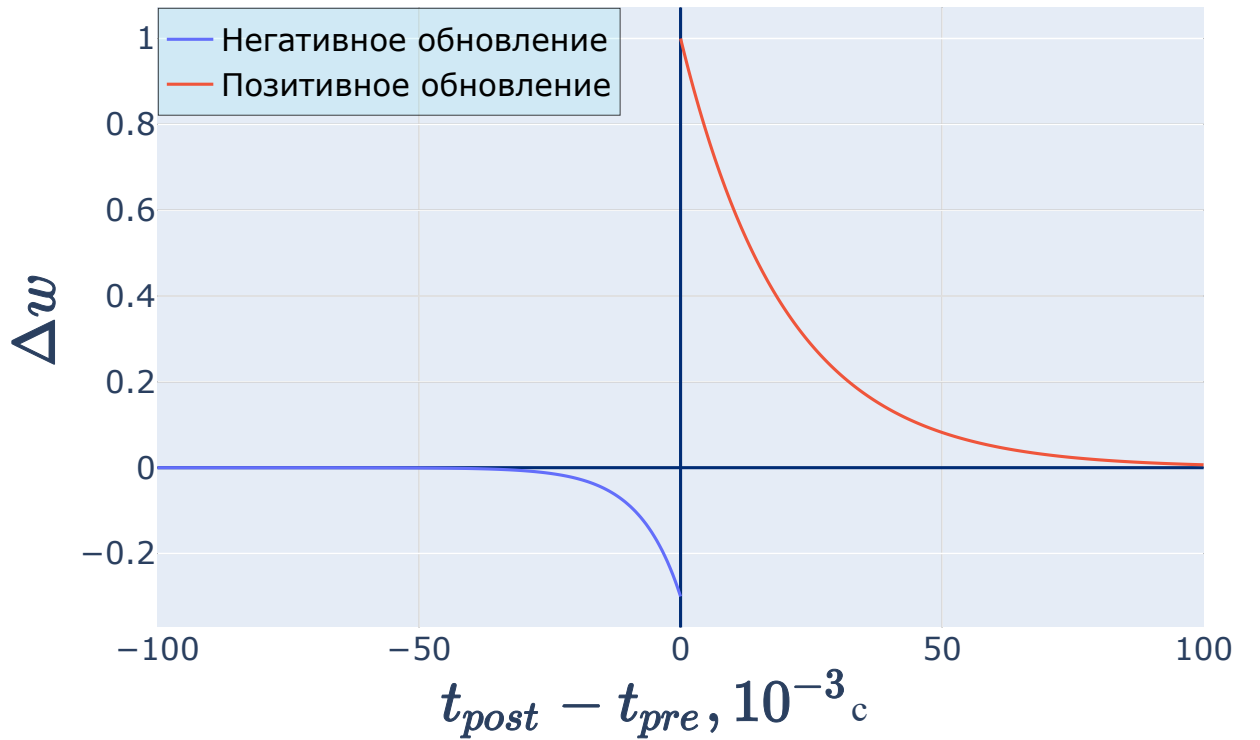


Рис. 7. Правило STDP. График зависимости изменения веса от разности времени регистрации пост- и пре-спайков.

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.

### 1.3.2 anti-STDP

anti-STDP является правилом обучения, обратным к STDP.

$$\Delta w = \begin{cases} A_+ \cdot e^{-\frac{t_{pre}-t_{post}}{\tau_+}}, & t_{pre} - t_{post} > 0 \\ A_- \cdot e^{-\frac{t_{pre}-t_{post}}{\tau_-}}, & t_{pre} - t_{post} < 0 \end{cases} \quad (8)$$

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

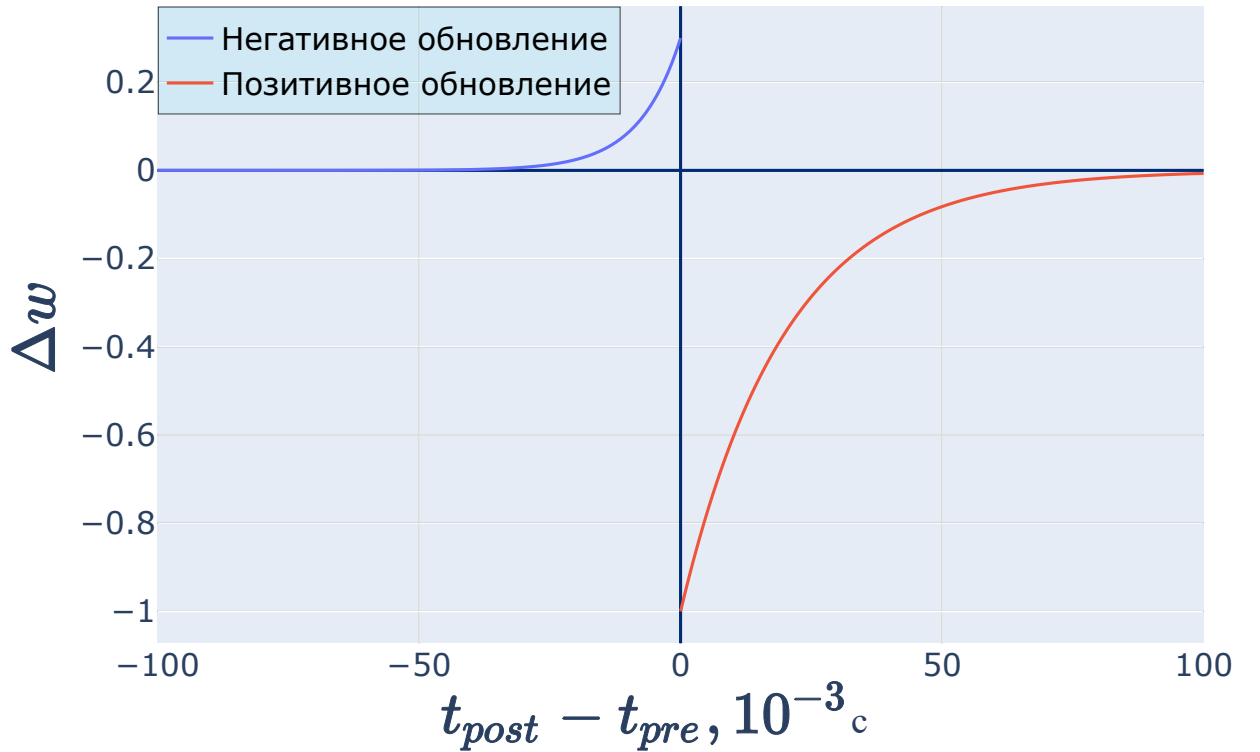


Рис. 8. Правило anti-STDP. График зависимости изменения веса от разности времени регистрации пост- и пре-спайков.



Таким образом, в процессе обучения вес связей между нейронами, которые активны одновременно, уменьшается. Это правило используется для обучения связей конкуренции.

## 1.4 Современные аппаратные реализации спайковых нейронных сетей

На сегодняшний день создано несколько нейроморфных процессоров для аппаратной реализации спайковых нейронных сетей. Рассмотрим самые крупные из них: 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 также указаны аналогичные параметры человеческого мозга.

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

Название	Нейроны	Синапсы	Энергопотребление, мВт
Мозг человека [18]	$\approx 16 \cdot 10^9$	$\approx 1 \cdot 10^{14}$	20000
TrueNorth [19]	$1 \cdot 10^6$	$256 \cdot 10^6$	$64^{(1)}$
SpiNNaker [20]	$1 \cdot 10^9$	$1 \cdot 10^{12}$	$1000^{(2)}$
Loihi [21]	$130 \cdot 10^3$	$130 \cdot 10^6$	неизвестно
Akida [22]	$1.2 \cdot 10^6$	$10 \cdot 10^9$	$\sim 10^{(3)}$

Все эти проекты разрабатываются под современные нужды и уже способны применяться в исследованиях или индустрии.

## Выводы к разделу 2

Выше были представлены несколько моделей нейронов в спайковых нейронных сетях и определены основные архитектуры, между которыми будет производиться сравнение. Показано, что аппаратная реализация спайковых нейронных сетей активно исследуется и уже существует в нескольких исполнениях. Локально соединенная архитектура имеет значительно меньшее количество параметров, чем полносвязная и в то же время позволяют обучать уникальный набор признаков для каждого рецептивного поля,

<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 мВт до 10 мВт в зависимости от сложности задачи.

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.

## 2 МОДЕЛИРОВАНИЕ ОБУЧЕНИЯ СПАЙКОВОЙ НЕЙРОННОЙ СЕТИ С КОНКУРЕНЦИЕЙ ЛОКАЛЬНЫХ РЕЦЕПТИВНЫХ ПОЛЕЙ

### 2.1 Описание задачи классификации

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 (2.4.1) обучающая выборка была разбита на 50000 изображений для обучения (обучающая выборка) и 10000 изображений для калибровки (калибровочная выборка).

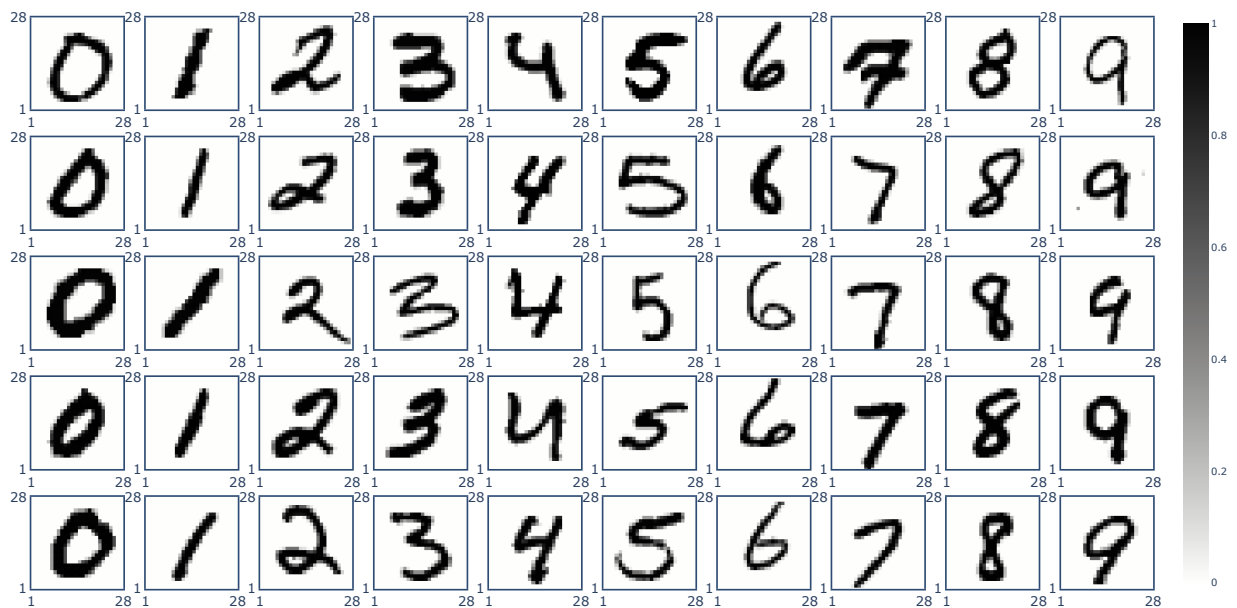


Рис. 9. Изображения MNIST

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

### 2.2 Особенности архитектуры

Нейросетевая архитектура LCSNN [16] вдохновлена строением зрительной коры мозга.  $Y$  слой сети состоит из  $n$  каналов, каждый из которых

является локально соединенным с  $X$  слоем нейронов.

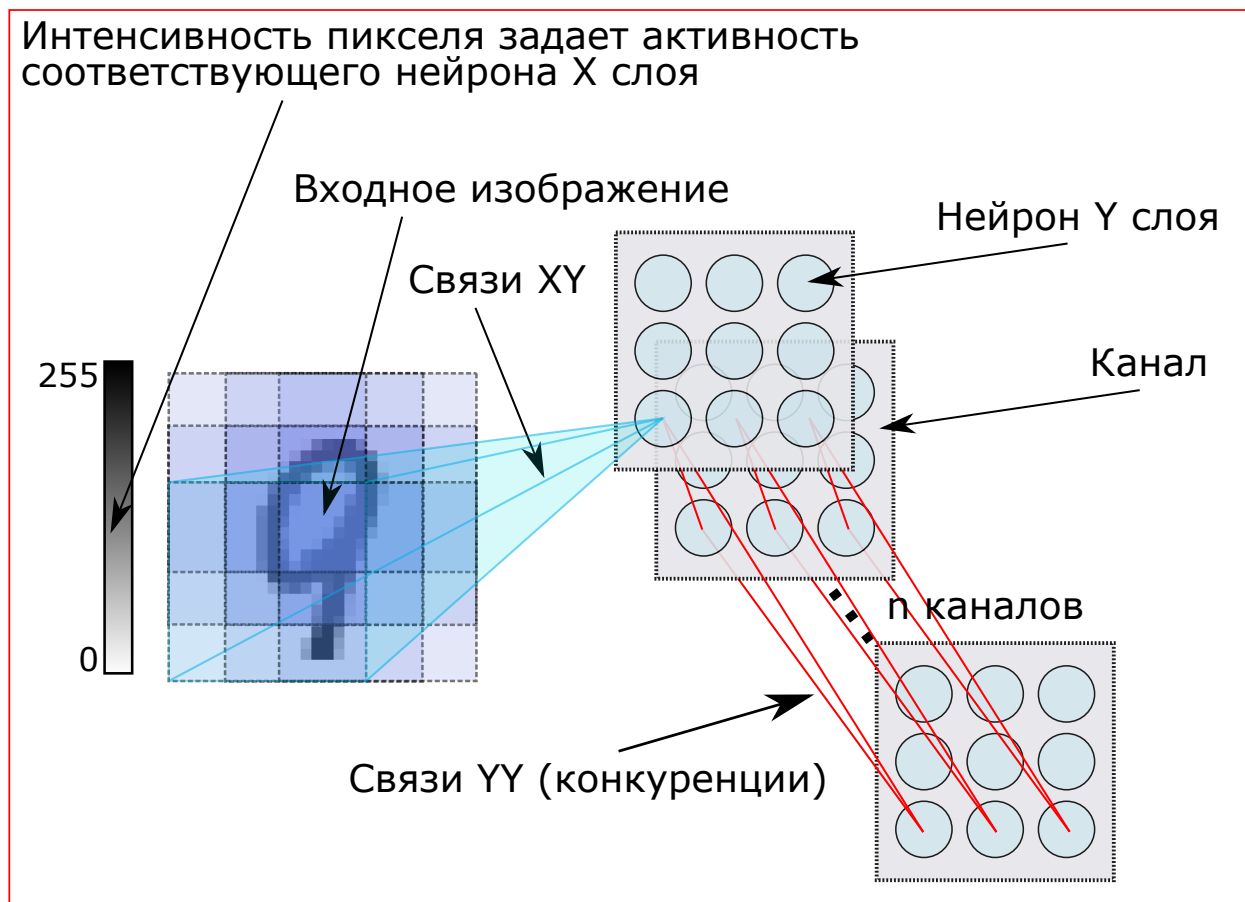


Рис. 10. Схема архитектуры LCSNN

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.

### 2.3 Обучение связей $XY$

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. 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.

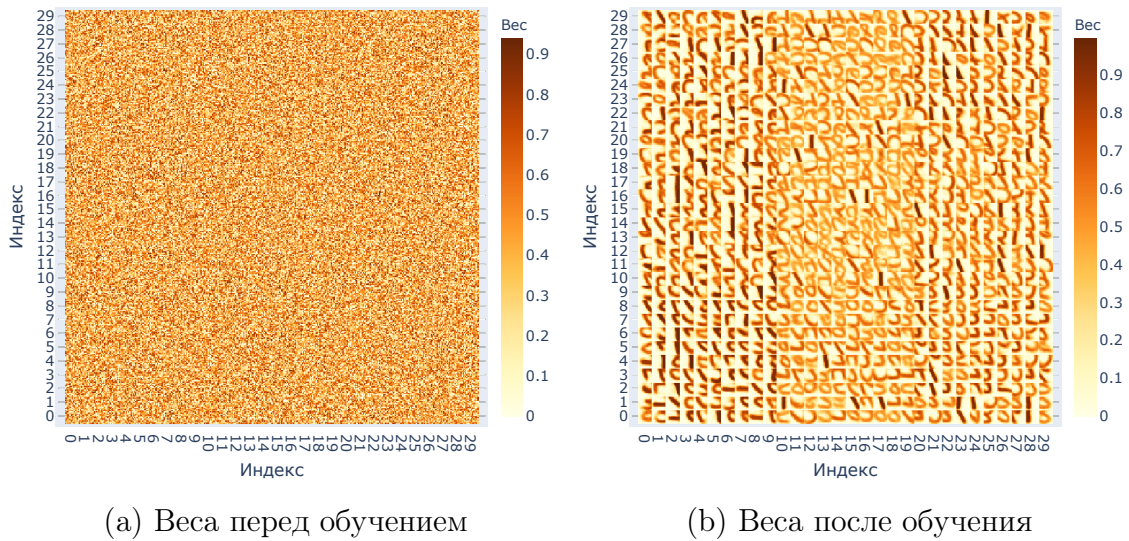


Рис. 11. Визуализация весов  $XU$  связей. Каждый квадратик  $12 \times 12$  соответствует весам одного  $U$  нейрона. Видно, что после обучения нейроны выучивают некоторые признаки, в которых явно угадываются элементы цифр.

## 2.4 Интерпретация активности спайковой нейронной сети

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

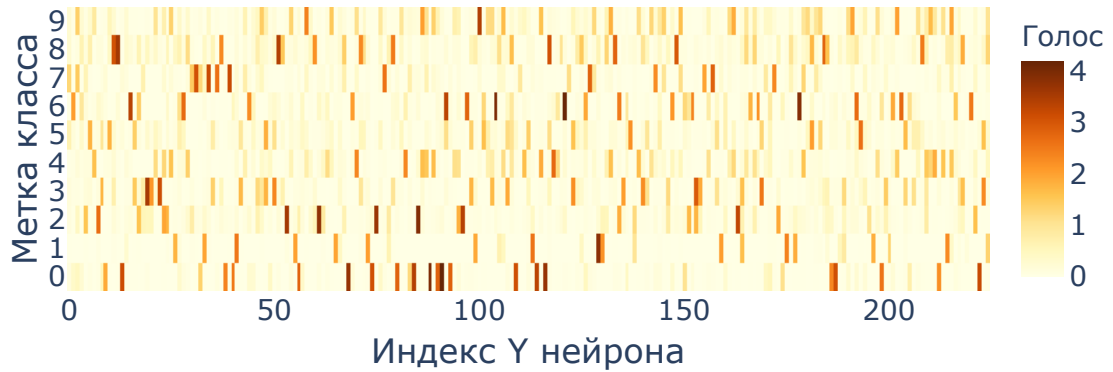
- Голосование патчей
- Общее голосование нейронов
- Голосование нейронов с предварительным отбором по спайкам
- Линейный классификатор

Преимущество первых трех методов заключается в их простоте. Они легко могут быть реализованы на аппаратном уровне. Однако, линейный классификатор значительно превосходит их по точности.

### 2.4.1 Калибровка голосов

Для первых трех методов необходимо произвести калибровку голосов нейронов. Каждому нейрону  $U$  слоя ставится в соответствие 10 чисел (голосов) для каждого возможного класса цифр (от 0 до 9). Голос вычисляется

как среднее число спайков данного нейрона в ответ на демонстрацию сети данной цифры. 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.



(a) Голоса нейронов.



(b) Среднее значение голоса для топ- $n$  класса, где топ-1 — класс с максимальным голосом нейрона, топ-10 — класс с минимальным голосом нейрона.

Рис. 12. 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 этих нейронов.

### Отбор по спайкам

Для каждого рецептивного поля ищется нейрон с максимальным количеством спайков. Ответом сети считается класс с максимальным результатом среди этих нейронов.

### Линейный классификатор

Суммы спайков нейронов  $Y$  слоя используются для обучения 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.

#### 2.4.2 Сравнение алгоритмов интерпретации активности сети

LCSNN отличаются большой скоростью обучения. 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 калибровки механизма интерпретации.

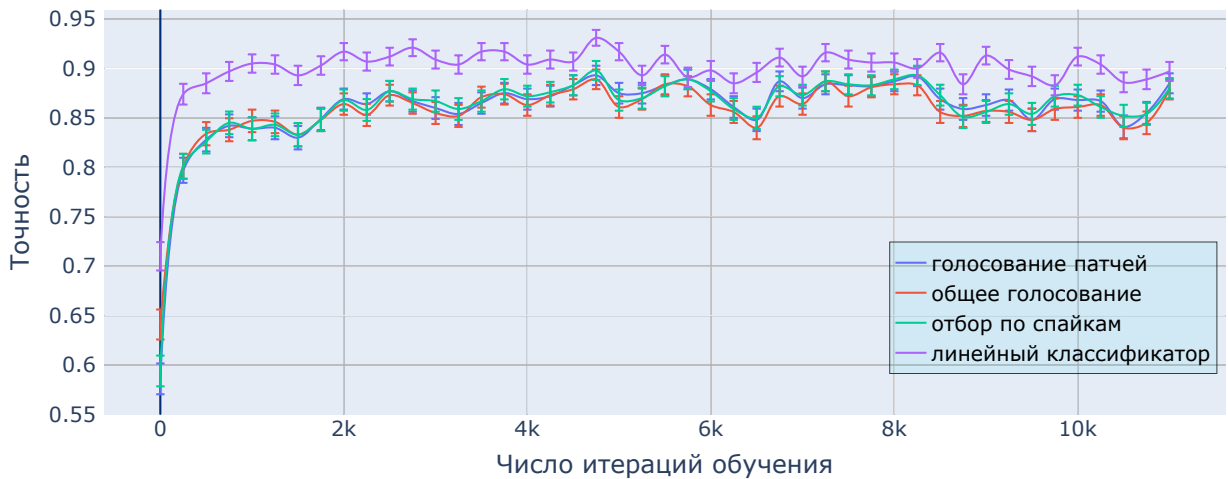


Рис. 13. Скорость обучения LCSNN. Сравнение различных алгоритмов интерпретации активности.

## 2.5 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.

Архитектура	Каналы	Ядро	Веса	Нейроны	Точность, %
LCSNN	1000	12	10287000	9000	$95.1 \pm 0.2$
LCSNN	100	12	218700	900	$87.5 \pm 0.3$
LCSNN	100	8	260800	1600	$82.9 \pm 0.4$
LCSNN <sup>(1)</sup>	25	12	37800	225	$82.3 \pm 0.4$
LCSNN	25	12	37800	225	$80.1 \pm 0.4$
LCSNN	25	8	35200	400	$73.6 \pm 0.4$
CSNN	169	12	255672	1521	$79.2 \pm 0.4$
CSNN	81	12	58464	729	$77.2 \pm 0.4$
CSNN	100	8	158464	1600	$77.4 \pm 0.4$
CSNN	25	12	5544	225	$65.8 \pm 0.5$
CSNN	25	8	9664	400	$63.1 \pm 0.5$
FCSNN	100	20	449100	100	$73.4 \pm 0.4$

Видно, что локально соединенная сеть превосходит сверточную сеть по точности даже при чуть превышающем числе параметров последней. Также видно, что не следует использовать слишком маленький 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.

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

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

---

<sup>(1)</sup>Сеть с обучением связей конкуренции

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

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

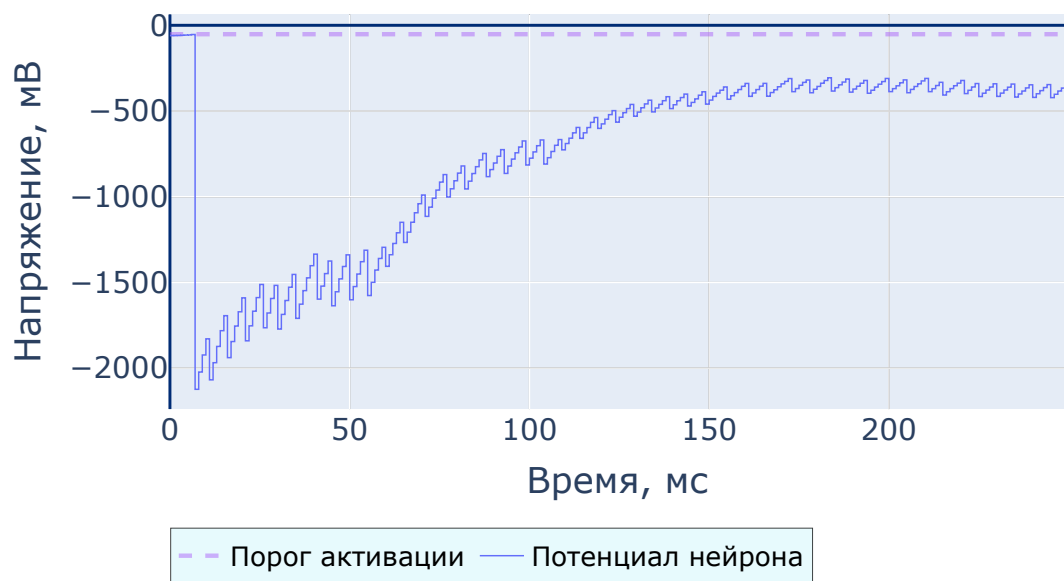
Результат, полученный в этой работе, практически соответствует результату из [16]. Заметим, что при помощи признаков, обученных без учителя достигаются результаты, лишь немногим уступающие результатам, полученным при помощи обучения с учителем. При этом лучшая локально соединенная сеть с конкуренцией локальных рецептивных полей из этой работы имеет  $10 \cdot 10^6$  связей (из них связей прямого распространения —  $1 \cdot 10^6$ , связей конкуренции —  $9 \cdot 10^6$ ), по сравнению с  $46 \cdot 10^6$  (из них связей прямого распространения —  $5 \cdot 10^6$ , связей конкуренции —  $41 \cdot 10^6$ ) у полносвязной сети с конкуренцией из [23]. К тому же, обучение сети из [23] велось  $1 \cdot 10^6$  итераций, тогда как в этой работе число итераций обучения не превосходит 5000.

## 2.6 Обучение связей $YU$ (конкуренции)

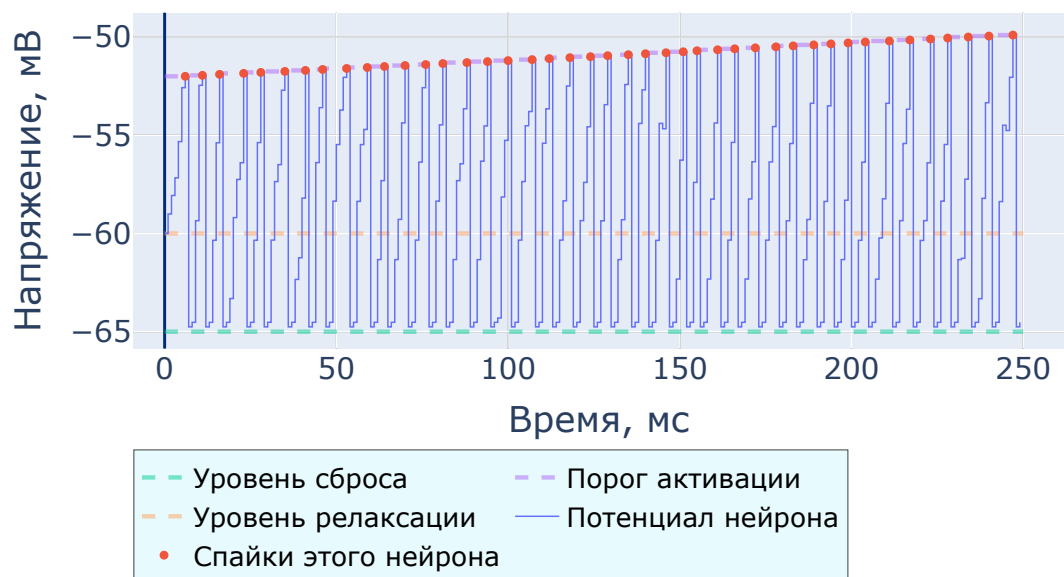
Связи  $YU$  очень сильно влияют на обучение связей  $XU$ . Большие по модулю значения способствуют вариативности в обучении  $Y$  нейронов, так для каждого рецептивного поля одновременно активными не могут быть нейроны, имеющие схожие веса  $XU$ . Малые по модулю значения не позволяют нейронам специализироваться.

Примечательно, что с такими большими по модулю значениями весов конкуренции для каждого рецептивного поля остается активным лишь один нейрон (или иногда несколько нейронов), веса которого лучше всего

соответствует области представленного изображения, который подавляет активность остальных.



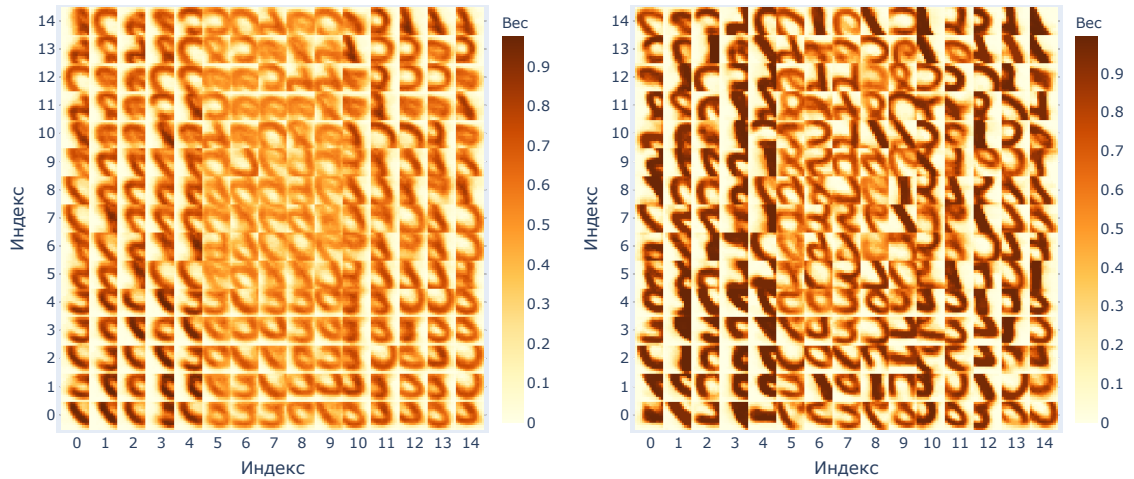
(а) Активность нейрона, который проиграл в конкуренции, подавлена другими нейронами.



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

Рис. 14. Влияние конкуренции на активность нейронов

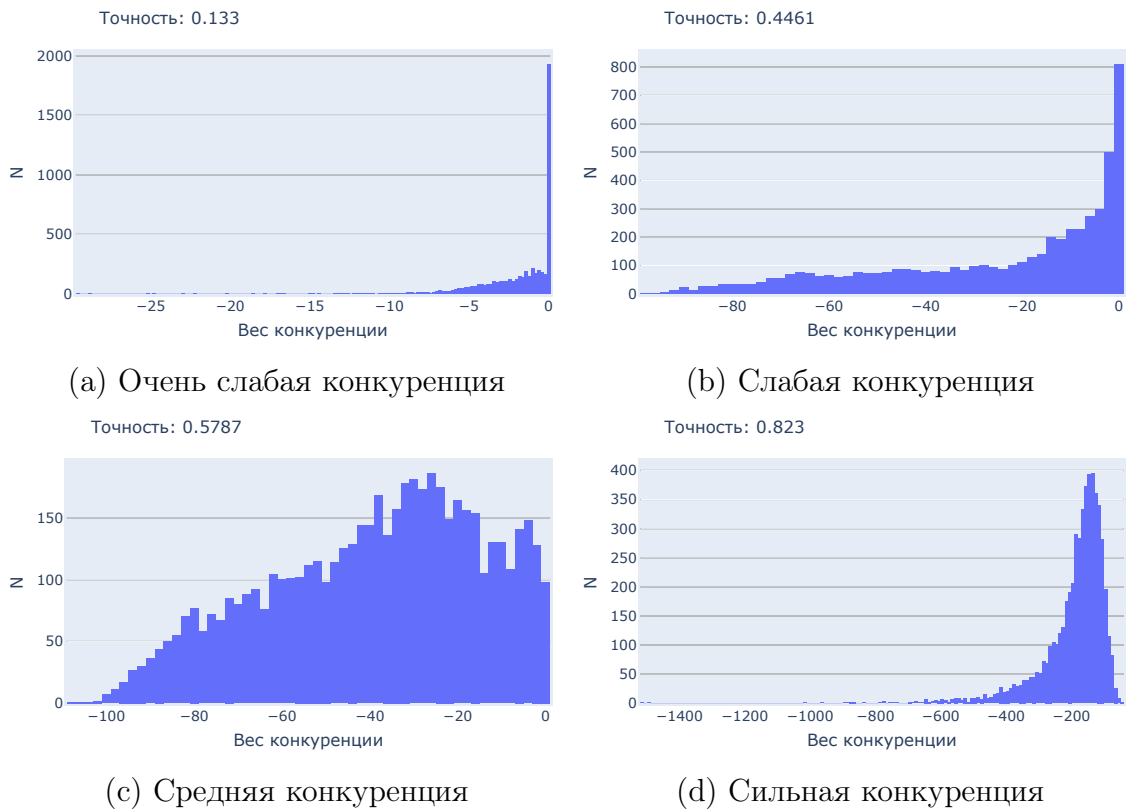
*Примечание: на графике слева не представлены значения  $v_{rest}$  и  $v_{reset}$ , так как на данном масштабе они практически совпадают с  $v_{thresh}$ .*



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

Рис. 15. 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 (разд. 1.3.2). При помощи варьирования параметров этого правила были получены различные распределения весов конкуренции.



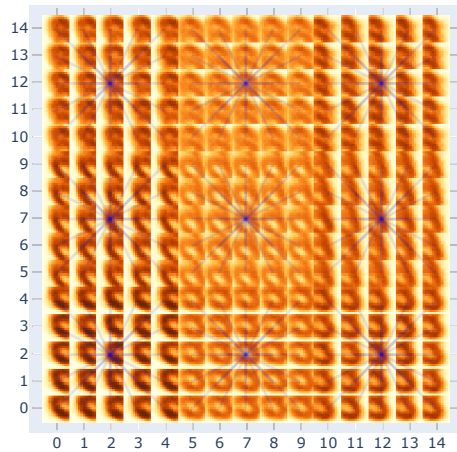
(а) Очень слабая конкуренция (б) Слабая конкуренция (в) Средняя конкуренция (д) Сильная конкуренция

Рис. 16. Различные распределения весов конкуренции

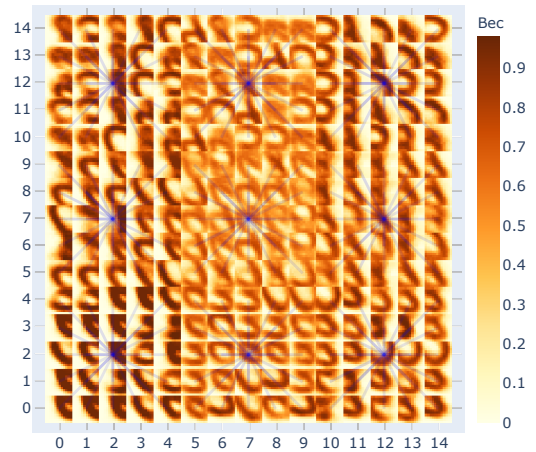
Видно, что точность сети повышается при тяготении распределения ве-

сов конкуренции в сторону больших по модулю значений. Заметим, что целью являлось не нахождение параметров сети, обеспечивающих максимальную точность, а исследования влияния обучения или не-обучения конкуренции на точность сети с данной конфигурацией остальных параметров.

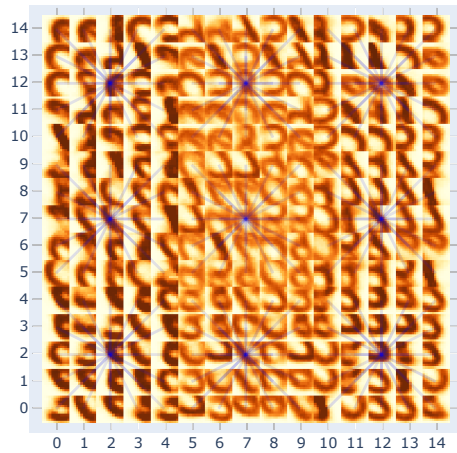
Примечательно, что не все связи  $YU$  получают большие по модулю значения. Это объясняется тем, что нейроны, специализирующиеся на значительном разных признаках, не нуждаются в конкуренции, так как они не проявляют высокую активность одновременно.



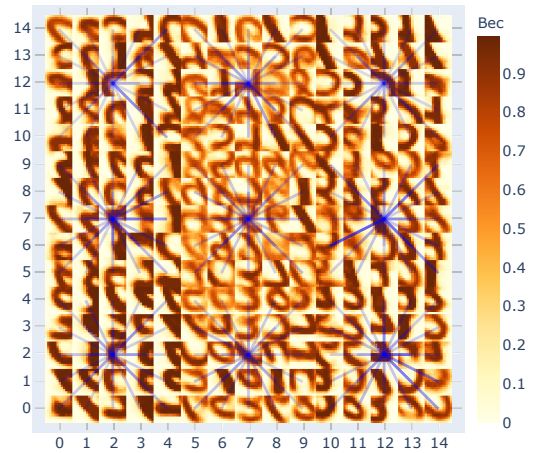
(a) Очень слабая конкуренция



(b) Слабая конкуренция



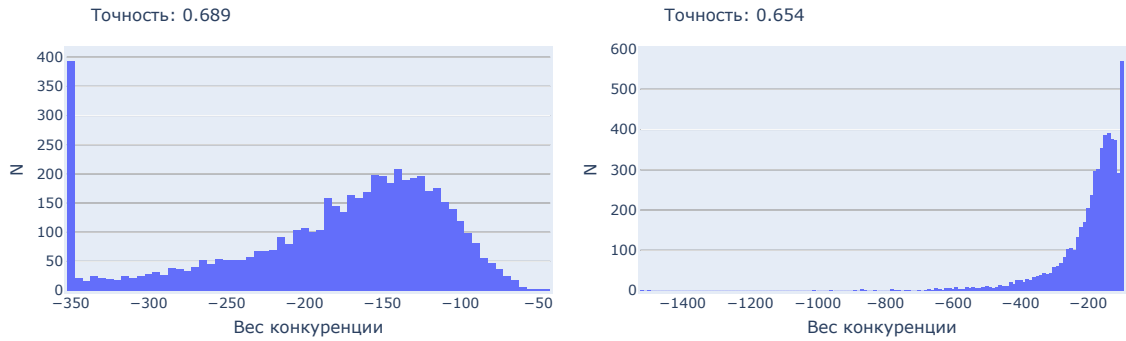
(c) Средняя конкуренция



(d) Сильная конкуренция

Рис. 17. Визуализация весов конкуренции поверх весов  $XU$ . Изображения соответствуют весам сетей с рисунка 16. 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 весов  $W_{ij}$  и  $W_{ji}$ ). Видно, что похожие признаки сильнее конкурируют между собой, чем различные.

Также были проведены эксперименты по ограничению значений весов конкуренции сверху и снизу. Оказалось, что как маленькие 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.



(а) Ограничение весов конкуренции снизу значением  $-350$  (б) Ограничение весов конкуренции сверху значением  $-100$

Рис. 18. Влияние ограничения значений весов конкуренции на точность. До ограничения точность этой сети составляла 0.823 (Рис. 16d).

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

### Выводы к разделу 3

В результате сравнения архитектур было показано, что локально соединенная архитектура превосходит сверточную архитектуру по точности при сравнимом числе параметров. Также было показано, что обучение связей конкуренции способно улучшить точность спайковой сети.

## ВЫВОДЫ

В результате проведения работы 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 нейросетевая архитектура, подходящая для реализации на 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.

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 изучение обучения конкуренции, а именно анализ распределений весов конкуренции, а также подбор оптимальных параметров обучения конкуренции. В этой работе не было исследовано обучение конкуренции у сетей с очень большим числом параметров и высокой точностью, так как такие исследования требуют слишком больших вычислительных ресурсов. Тем не менее, возможно, что обучение конкуренции позволит улучшить точность для таких сетей и превзойти текущие state of the art результаты.



## БЛАГОДАРНОСТИ

Выражаю благодарности Дёмину Вячеславу Александровичу за поддержку и руководство исследовательским процессом, Нехаеву Дмитрию Вадимовичу за терпеливую и кропотливую работу со мной в течение всего года, а также Королёвой Александре Валерьевне за помощь со стороны Физического факультета МГУ.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. *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>.
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. 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>.
4. Enabling Spike-Based Backpropagation for Training Deep Neural Network Architectures / C. Lee [и др.] // *Frontiers in Neuroscience*. — 2020. — февр. — т. 14. — ISSN 1662-453X. — DOI: [10.3389/fnins.2020.00119](https://doi.org/10.3389/fnins.2020.00119). — URL: <http://dx.doi.org/10.3389/fnins.2020.00119>.
5. *Lee J., Delbruck T., Pfeiffer M.* Training Deep Spiking Neural Networks Using Backpropagation // *Frontiers in Neuroscience*. — 2016. — авг. — т. 10. — DOI: [10.3389/fnins.2016.00508](https://doi.org/10.3389/fnins.2016.00508).
6. Regularization of Neural Networks using DropConnect / L. Wan [и др.]. — 2013. — июнь. — URL: <http://proceedings.mlr.press/v28/wan13.html>.
7. *Edwards C.* Growing pains for deep learning // *Commun. ACM*. — 2015. — т. 58. — с. 14—16.
8. 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).
9. *Pehlevan C.* A Spiking Neural Network with Local Learning Rules Derived From Nonnegative Similarity Matching. — 2019. — arXiv: [1902.01429 \[cs.NE\]](https://arxiv.org/abs/1902.01429).
10. *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>.

11. *Tang G., Shah A., Michmizos K. P.* Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM. — 2019. — arXiv: [1903.02504 \[cs.R0\]](#).
12. *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](#).
13. *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](#).
14. 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](#). — URL: <http://dx.doi.org/10.1145/3304103>.
15. Gradient-based learning applied to document recognition / Y. Lecun [и др.] // *Proceedings of the IEEE*. — 1998. — т. 86, № 11. — с. 2278–2324.
16. Locally Connected Spiking Neural Networks for Unsupervised Feature Learning / D. J. Saunders [и др.]. — 2019. — arXiv: [1904.06269 \[cs.NE\]](#).
17. *Abusnaina A., Abdullah R.* Spiking Neuron Models: A Review // *International Journal of Digital Content Technology and its Applications*. — 2014. — июнь. — т. 8. — с. 14–21.
18. 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](#). — eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cne.21974>. — URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cne.21974>.
19. 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](#). — eprint: <https://science.sciencemag.org/content/345/6197/668.full.pdf>. — URL: <https://science.sciencemag.org/content/345/6197/668>.
20. 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](#).

21. 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).
22. Akida Neural Processor System-on-Chip. — 2020. — апр. — URL: <https://brainchipinc.com/akida-neuromorphic-system-on-chip/>.
23. *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>.
24. *Tavanaei A., Maida A. S.* Multi-layer unsupervised learning in a spiking convolutional neural network. — 2017.
25. 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>.
26. *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.

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