

Обучение связей конкуренции локальных рецептивных полей в спайковых нейронных сетях

Даниил Гафни, Дмитрий Нехаев, Вячеслав Дёмин

Аннотация

Стандартные методы обучения (на основе метода обратного распространения ошибки), применяющиеся в формальных искусственных нейронных сетях (ИНС) затруднительно применять к спайковым нейронным сетям (СНС) из-за дискретной и распределенной во времени природы генерируемых ими событий – импульсов, или спайков. 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 ИНС 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, например, таких как синаптическая пластиность, зависящая от моментов импульсов STDP, для использования в различных архитектурах СНС является важной задачей.

Введение связей конкуренции (имеющих отрицательные веса) позволяет добиться лучшего разделения выучиваемых нейронами признаков. Самым распространенным способом введения конкуренции нейронов СНС является использование группы методов «победитель забирает все», когда первый нейрон в слое, который генерирует импульс, либо обнуляет мембранные переменные других нейронов, либо накладывает запрет на генерацию другими нейронами спайков в

течение некоторого заданного времени. Использование связей конкуренции в виде отрицательных синаптических весов изучено гораздо хуже, а влияние возможности их обновления по локальным правилам, наряду с возбуждающими проекциями, практически не исследовалось. В настоящей работе показано, что обучение связей конкуренции в архитектуре локально связной СНС потенциально позволяет добиться более хороших результатов в точности распознавания образов (цифр MNIST) за счет организации конкурентной и одновременно кооперативной работы групп нейронов.

Введение

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 десятков [2; 4] и даже сотен миллиардов (последняя языковая модель OpenAI GPT-3 [5]).

Спайковые нейронные сети (СНС) являются альтернативным перспективным нейроморфным алгоритмом и имеют ряд преимуществ перед формальными нейронными сетями.

- С помощью СНС возможно более богатое динамичное кодирование паттернов в непрерывном времени, которое, как правило, недоступно для формальных нейронных сетей [6]. В связи с этим, наибольший интерес СНС представляют для решения задач с временными последовательностями данных (обработка видеопотоков, звуковых рядов, речи, принятие решений, управление).
- Для обучения СНС могут применять локальные алгоритмы [7—9], 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 СНС 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.
- СНС могут быть реализованы на специализированных нейроморфных процессорах (как на базе цифровых элементов [10—12], так и с использованием гибридных цифро-аналоговых схем (BrainScales, [13])), в том числе на основе мемристоров [ссылки на чужие (Ielmini, Querlioz) и наши работы]), обладающих сверхнизким энергопотреблением (достигается за счет сокращения количества актов и средней длины передачи сигналов, а также пространственной и временной разреженности спайков в сети). Такая реализация вместе с чисто алгоритмическим преимуществом СНС дает также существенный выигрыш в производительности. [14; 15].
- СНС в большей степени, чем формальные нейронные сети, биологически корректно моделируют взаимодействия нейронов, что может использоваться для биологических симуляций нервной системы в целях не только биоинформационических, но также биофизических и медицинских аспектов функционирования.

Представляет интерес изучение на базе СНС известных архитектур ИНС, таких как сверточная, локально соединенная и полносвязная архитектуры [2]. Более того, введение дополнительных рекуррентных связей конкуренции в этих топологиях способствует лучшему разделению признаков в процессе обучения [16; 17]. 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 при формировании отдельной карты признаков. Локально соединенная архитектура имеет значительно меньшее количество параметров, чем полносвязная, и 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 (пространственно скоррелированных данных). Наличие конкуренции в СНС позволяет делать их признаки более независимыми. Поэтому для экспериментов в настоящей работе были выбраны локально соединенные СНС со связями конкуренции, а в качестве базовых моделей — сверточная и полносвязная сети со связями конкуренции, а именно:

- i изучается влияние обучения связей конкуренции [16–18] между нейронами на точность распознавания образов в задаче классификации рукописных изображений цифр из датасета MNIST [19] при обучении без учителя для архитектуры локально соединенной сети (Locally Connected Spiking Neural Network, LCSNN) [20];
- ii проводится сравнение этой архитектуры со сверточной сетью (Convolution Spiking Neural Network, CSNN) и полносвязной сетью (Fully Connected Spiking Neural Network, FCSNN).

1 Обучение СНС с необучаемой конкуренцией локальных рецептивных полей

1.1 Описание задачи

Для сравнительного анализа 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 28×28 пикселя и являются черно-белыми. Из-за необходимости калибровки сетей (см. ниже) обучающая выборка была разбита на 50000 изображений для обучения (обучающая выборка) и 10000 изображений для калибровки (калибровочная выборка).

В этой работе изображения обрезаются так, что используется только центральная область 20×20 пикселей.

1.2 Особенности архитектуры сетей с конкуренцией

В данной работе изучаются сети с использованием локально соединенных, полносвязных и сверточных слоев. В полносвязном слое каждый нейрон соединен с каждым нейроном из предыдущего слоя. В сверточном слое 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]. Архитектура локально соединенного слоя совпадает с таковой для сверточного слоя, 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 синаптическими связями с отрицательным весом. Так как общий вид этих сетей совпадает, рассмотрим его на примере СНС с локально соединенным слоем.

Интенсивность пикселя задает активность соответствующего нейрона X слоя

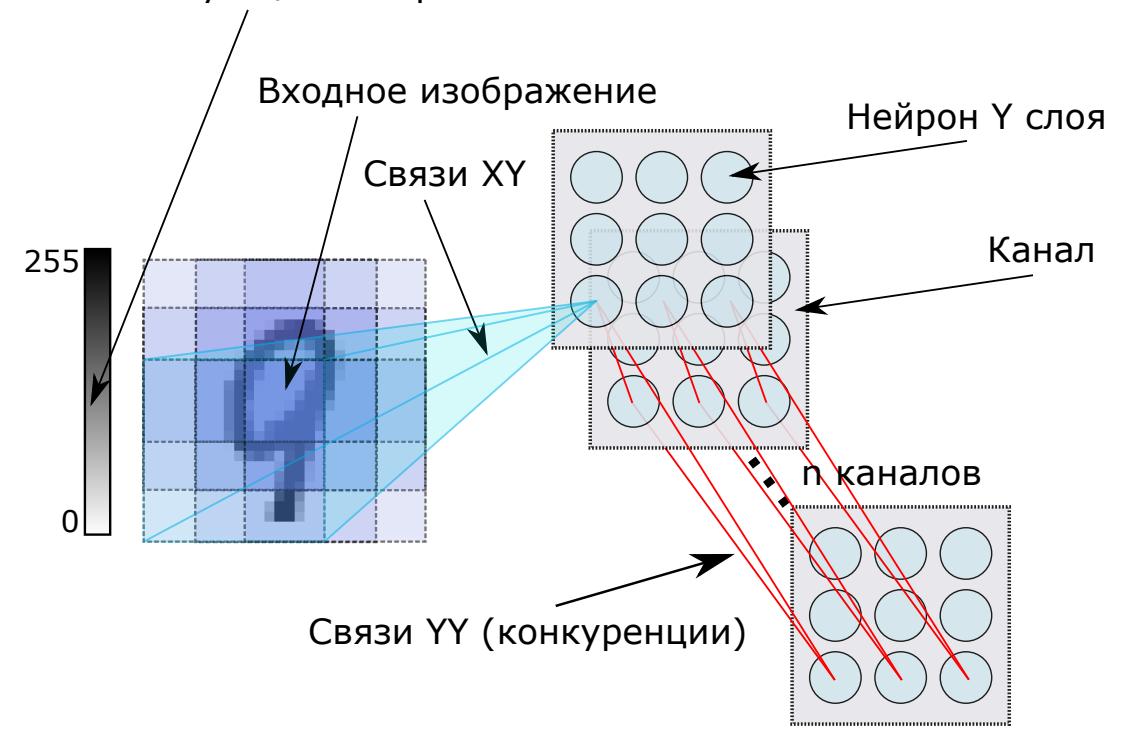


Рис. 1. Схема архитектуры LCSNN

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.

Нейроны, имеющие общие рецептивные поля (патчи), соединяются YY связями конкуренции. Такие связи имеют отрицательные веса, а значит, негативно влияют на активность. Нейроны, не имеющие общего патча (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. Исследовались сети с числом Y нейронов $\approx 100 - 900$ и параметрами свертки: ширина - 8 или 12, сдвиг - 4. Так как изображения MNIST обрезались до размера 20×20 , то число количества нейронов на канал составляло

3 или 4.

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, τ_v — временная константа симуляции, а R — размерный коэффициент, численно равный 1.

Порог активации v_{thresh} у ALIF нейрона не является константой, а немного повышается при каждом спайке, релаксируя затем к своему начальному значению θ_0 . Динамика порога активации задается следующими уравнениями:

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

где θ_0 — начальный порог активации, $\theta(t)$ — адаптивная добавка к порогу активации нейрона после генерации каждого спайка, которая вычисляется из условия

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

После генерации каждого спайка наступает короткий период рефрактерности, при котором на протяжении времени $t_{refract}$ потенциал нейрона остается на уровне v_{reset} . Начальные веса связей задаются случайными числами из равномерного распределения.

1.3 Обучение прямых связей

Для уменьшения числа параметров модели изображения обрезаются до размера 20×20 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 [7]. Это биологически инспирированное правило обучения без учителя [7]. При получении пресинаптического импульса (пре-спайка) и испускании пост-спайка вес w связи, по которой пришел пре-спайк, увеличивается на Δw , где

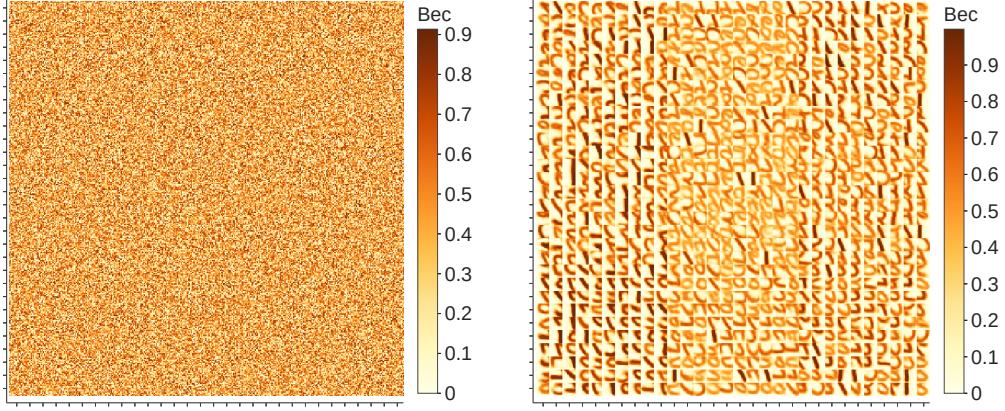
$$\Delta w = \begin{cases} A_+ \cdot e^{-\frac{t_{pre}-t_{post}}{\tau_+}}, & t_{pre} - t_{post} > 0 \\ A_- \cdot e^{-\frac{t_{pre}-t_{post}}{\tau_-}}, & t_{pre} - t_{post} < 0 \end{cases} \quad (4)$$

Заметим, что

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

Таким образом, в процессе обучения у каждого нейрона увеличивается вес связей, по которым пре-спайк систематически приходит непосредственно 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 генерировать пост-спайк, если в некотором коротком промежутке времени эти нейроны будут активны одновременно (что означает скоррелированность активностей данных нейронов при выделении некоторого признака). Обратное правило (с противоположными по знаку A_+ и A_-) называется правилом anti-STDP [22]. 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 равна определенной константе. Это делается для избежания возникновения слишком больших отдельных весов. Значение



(а) Перед обучением

(б) После 10000 итераций обучения

Рис. 2. Визуализация весов XY связей сети с 900 Y нейронами (по 100 нейронов на каждый из 9 патчей) - матрица 360×360 . Веса одного нейрона соответствуют квадрату 12×12 . Веса Y нейронов сгруппированы по патчам - левый нижний квадрат 120×120 (10×10 нейронов) соответствует левому нижнему патчу, центральный квадрат соответствует центральному патчу, и так далее. Видно, что после обучения нейроны выучивают некоторые признаки, в которых явно угадываются элементы цифр.

константы нормализации есть важный гиперпараметр модели, который подбирается для каждой конкретной архитектуры.

1.4 Интерпретация активности спайковой нейронной сети

Для интерпретации активности нейронов Y слоя (то есть соотнесения их активности с определенным классом распознаваемых цифр) использовалось несколько методов: голосование патчей, общее голосование нейронов, голосование нейронов с предварительным отбором по спайкам или линейный классификатор. Преимущество первых трех методов заключается в их простоте. Однако, линейный классификатор значительно превосходит их по точности.

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 калибровочной симуляции) —

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

где s_t принимает значение 0 или 1 в зависимости от наличия спайка в данный момент времени, n — размер калибровочной выборки для каждой цифры ($n = 1000$).

Для всех сетей использовалась калибровка на 10000 примерах из калибровочной выборки, специально отобранный из тестового набора данных, как было отмечено выше. Заметим, что калибровка не является частью обучения сети, так как она входит лишь в алгоритм интерпретации поведения СНС. Эти голоса используются как мера уверенности нейрона в каждом из классов.

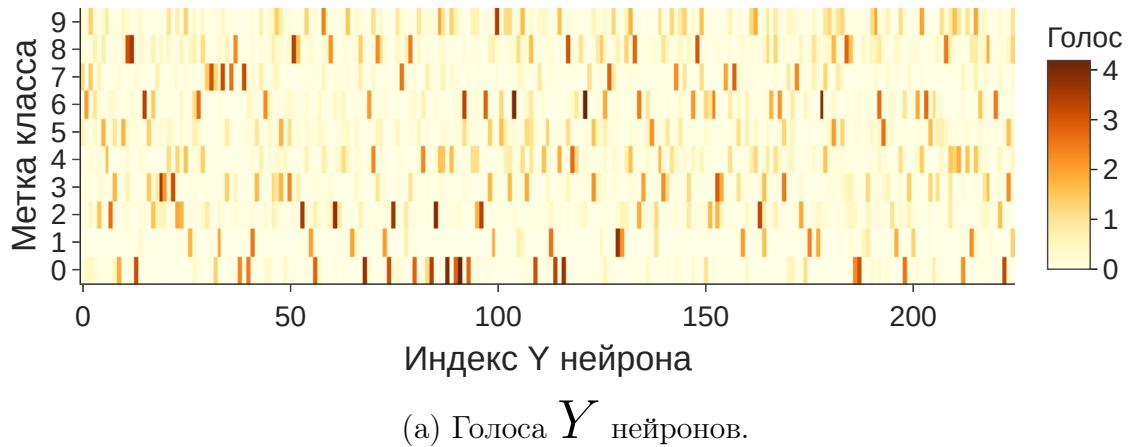
Далее результатом будем называть произведение количества спайков нейрона на голос.

При общем голосовании ответом сети считается класс с максимальным результатом среди всех нейронов.

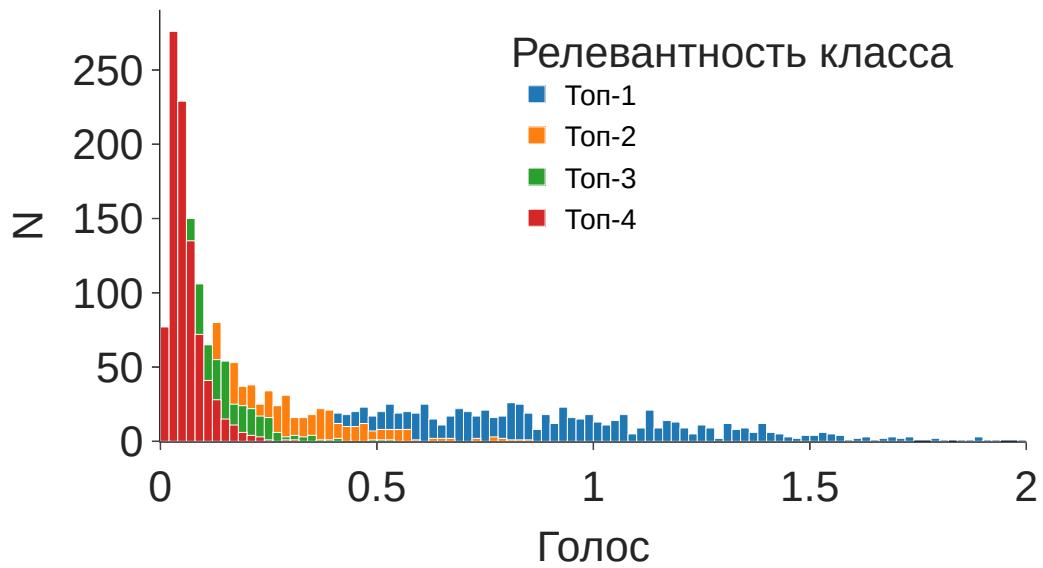
При голосовании патчей для каждого рецептивного поля ищется нейрон с максимальным результатом. Ответом сети считается класс с максимальным результатом среди таких нейронов для всех рецептивных полей.

При отборе 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.

При использовании линейного классификатора на его вход подаются суммы спайков отдельных нейронов Y слоя, а в качестве целевой переменной используются истинные метки классов изображений.



(a) Голоса Y нейронов.



(b) Распределения голосов нейронов по четырем наиболее релевантным классам (для каждого нейрона). Видно, что величина голоса (среднее число спайков на класс) сильно падает с уменьшением релевантности.

Рис. 3. Визуализация голосов нейронов Y слоя сети с 225 Y нейронами. Высокие значения соответствуют большой специализации нейрона на классе. Видно, что практически все нейроны имеют отличное от нуля значение голоса, то есть неактивных нейронов в сети практически нет.

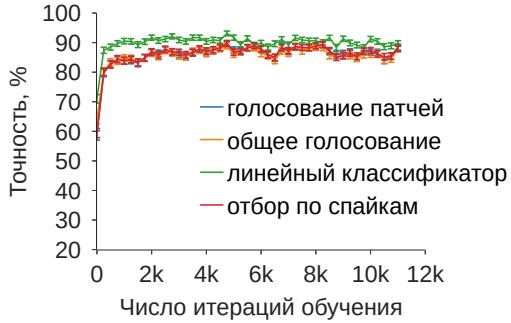
Как и в других методах интерпретации активности сети, обучение ведется на калибровочной выборке.

Для оценки работы алгоритма интерпретации используется точность — отношение количества верно распознанных объектов выборки к размеру тестовой выборки. В этой работе размер тестовой выборки (при измерении точности для отдельных сетей) составляет 10000.

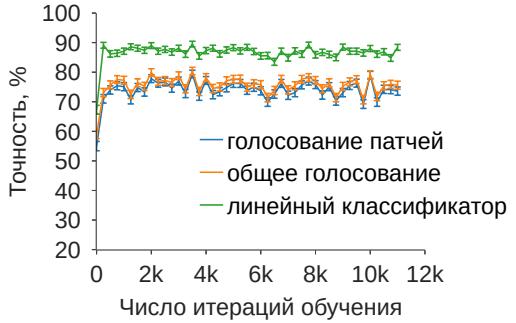
С целью сравнения алгоритмов интерпретации активности сети были построены кривые обучения для различных алгоритмов интерпретации. Точность измерялась через каждые 250 итераций обучения. Для калибровки алгоритма интерпретации в каждой точке использовалась калибровочная выборка объемом 5000, тогда как для измерения точности — тестовая выборка объемом 1000.

Видно, что через несколько тысяч итераций точность распознавания выходит на плато насыщения, после чего уже не возрастает. Также заметно, что три метода голосования в целом не отличаются по точности, тогда как линейный классификатор значительно превосходит их все. Следует отметить, что точность даже необученной сети может достигать 70% за счет того, что даже при случайной 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 полно связную по скорости обучения, так как при каждой итерации обучения обновляется большее число параметров (для каждого рецептивного поля активен минимум один нейрон). В настоящем исследовании не наблюдается такого эффекта, так как представленные кривые обучения построены для недостаточно больших сетей. Интересно, что использование линейного классификатора не повышает точность распознавания для полно связной сети — скорее всего, из-за выделения признаков для голосования по всему изображению, в отличие от сверточной и локально связной архитектур.

Линейный классификатор превосходит LCSNN- и CSNN-based алгоритмы голосования по точности, так как является обобщением голосования в том смысле, что при его работе также используется



(a) Кривая обучения LCSNN.

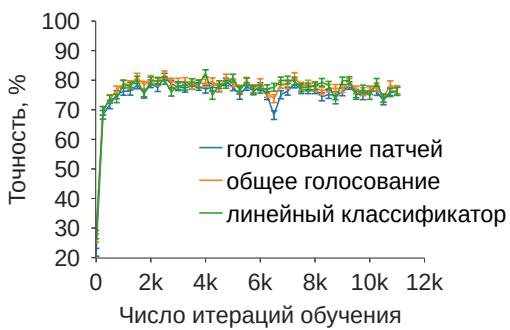


(b) Кривая обучения CSNN.

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.

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 LCSNN связи конкуренции. Из-за высоких вычислительных нагрузок не ставилось задачи по нахождению параметров, обеспечивающих максимальную точность для каждой архитектуры. Эти параметры были подобраны приблизительно, однако, судя по результатам отдельных отладочных экспериментов, величина расхождения с оптимальными значениями не превышает 1-2%. 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.



(c) Кривая обучения FCSNN.

Рис. 4. Сравнение кривых обучения различных архитектур спайковых нейронных сетей с конкуренцией: (4a) LCSNN, (4b) CSNN, (4c) FCSNN. 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. Сети имеют 100 каналов (для FCSNN канал состоит из единственного нейрона), размер свертки для LCSNN составляет 12, а для CSNN — 8.

Таблица 1. Результаты сравнения различных архитектур спайковых нейронных сетей. Для каждой конфигурации точность измерялась $N = 5$ раз. В таблице указаны среднее арифметическое значение точности и его стандартное отклонение. «Ядро» соответствует числу k в размере свертки $k \times k$.

N	Конфигурация					Точность, %	
	Архитектура	Каналы	Ядро	Параметры	Y нейроны	Метод ⁽¹⁾	Метод ⁽²⁾
1	LCSNN	1000	12	10287000	9000	92.3 ± 0.7	95.1 ± 0.5
2	LCSNN	100	12	218700	900	87.5 ± 0.9	91.5 ± 0.6
3	LCSNN	100	8	260800	1600	82.9 ± 0.6	88.1 ± 0.7
4	LCSNN ⁽³⁾	25	12	37800	225	82.3 ± 1.0	88.2 ± 0.6
5	LCSNN	25	12	37800	225	80.1 ± 1.0	85.5 ± 0.8
6	LCSNN	25	8	35200	400	73.6 ± 1.0	80.3 ± 0.7
7	CSNN	169	12	279864	1521	79.2 ± 1.6	85.7 ± 1.4
8	CSNN	81	12	69984	729	77.2 ± 1.7	83.1 ± 1.2
9	CSNN	100	8	164800	1600	77.4 ± 1.9	82.1 ± 1.3
10	CSNN	25	12	9000	225	65.8 ± 0.7	77.1 ± 0.6
11	CSNN	25	8	11200	400	63.1 ± 1.2	75.8 ± 0.5
12	FCSNN	100	20	49900	100	81.4 ± 0.9	82.1 ± 0.8

Видно (№2 и №7 в таблице 1), что локально соединенная сеть превосходит сверточную сеть по точности даже при чуть превышающем числе параметров последней. Также, можно заметить, что не следует использовать слишком малый размер ядра свертки. Это приводит к выучиванию менее существенных признаков. Действительно, в традиционном машинном обучении на основе ИНН используются либо неглубокие сети с большими ядрами свертки, либо глубокие модели, но, наоборот, с маленькими ядрами свертки.

Следует отметить, что SNN могут достигать значительно больших точностей распознавания на MNIST. Для этого можно использовать: (i) сети с существенно большим числом весовых параметров (например, с увеличенным числом каналов); (ii) более глубокие сети (с большим количеством слоев); (iii) более эффективные на сегодня алгоритмы обучения (например, обучение с учителем или с использованием контрастивной функции потерь [23]). В то же время, достижение максимально возможной точности не являлось целью настоящего исследования, адресованного, в первую очередь, сравнению различных архитектур SNN.

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 [24]. К тому же, обучение сети из [24] велось в течение $1 \cdot 10^6$ итераций, тогда как в этой работе число итераций обучения не превосходит 5000.

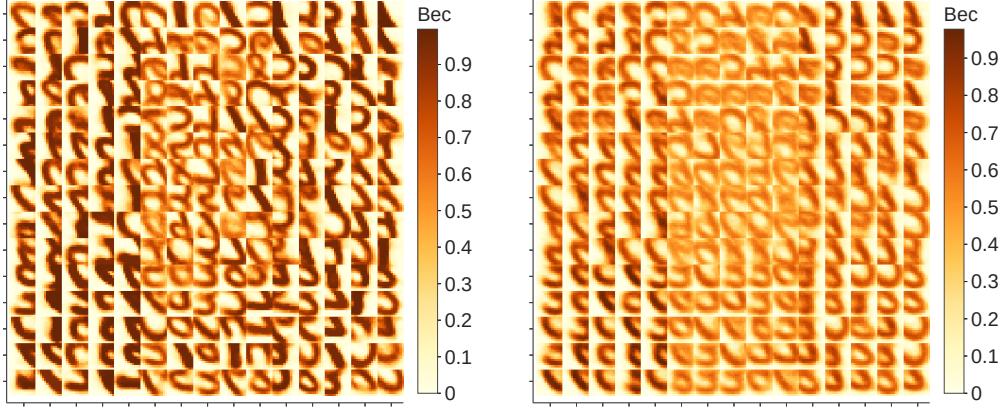
⁽¹⁾Лучший алгоритм голосования

⁽²⁾Линейный классификатор

⁽³⁾Сеть с обучением связей конкуренции

Таблица 2. Результаты других исследований спайковых нейронных сетей. Во всех используется датасет MNIST.

Статья	Архитектура	Обучение	Точность, %
Эта работа	Локальная + конкуренция	Без учителя	95.1 ± 0.5
[20]	Локальная + конкуренция	Без учителя	95.07 ± 0.63
[24]	Полносвязная + конкуренция	Без учителя	95
[16]	Полносвязная + конкуренция	С учителем / с частичным привлечением учителя	95.4 / 72.1
[25]	Сверточная	С частичным привлечением учителя	96.95 ± 0.08
[26]	Сверточная	С частичным привлечением учителя	99.28 ± 0.10
[27]	Сверточная	С частичным привлечением учителя	97.20 ± 0.07



(a) Высоко специализированные веса, вес конкуренции равен -100 . (b) Слабо специализированные веса, вес конкуренции равен -10 .

Рис. 5. Влияние конкуренции на обучение связей XY .

2 Обучение связей конкуренции

Ингибирующие связи YY существенно влияют на обучение связей прямого распространения сигнала XY . Большие по модулю значения весов конкуренции способствуют большей вариативности и специализации в обучении Y нейронов, так как для каждого рецептивного поля одновременно активными не могут быть нейроны, имеющие схожие веса XY (Рис. 5а). Наоборот, малые по модулю веса конкуренции не позволяют нейронам эффективно специализироваться (Рис. 5а).

Все SNN, о которых шла речь в настоящей работе до этого момента, имели фиксированные веса конкуренции. В связи с этим, возникает вопрос, как влияет на точность сети обучение весов конкуренции. Для ответа на поставленный вопрос с целью обновления связей YY было выбрано правило anti-STDP (правило, противоположное по знакам A_+ и A_- в стандартном STDP). При варьировании значений параметров этого правила были получены различные распределения весов конкуренции. Начальные значения весов конкуренции задавались из равномерного распределения от 0 до некоторого числа, которое в дальнешем будем называть начальным весом конкуренции.

Таблица 3. Параметры anti-STDP

Рисунок	τ_+ , мс	τ_- , мс	A_+	A_-
7a	14.7	14.2	-0.5	-1.5
7b	5.4	15.1	-1.2	-0.6
7c	17.6	24.5	-1.9	-1.6
7d	17.7	16.5	-0.1	-1.6

Эксперименты проводились с LCSNN.

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

Примечательно, что не все связи YY получают большие по модулю значения (Рис. 6d). Это объясняется тем, что нейроны, специализирующиеся на существенно разных признаках, не нуждаются в конкуренции, так как они не проявляют высокую активность одновременно.

Обучение конкуренции производилось только на сетях из 25 каналов (225 нейронов), так как его моделирование требует больших вычислительных ресурсов.

В ходе оптимизации моделей была измерена точность большого количества LCSNN с различными конфигурациями гиперпараметров (Рис. 8). Видно, что конфигурации с высокими точностями (темные ломаные) локализуются определенным образом - дописать

2.1 Анализ обученных весов конкуренции

Дополнительно были проведены эксперименты по ограничению значений весов конкуренции по модулю сверху и снизу до проведения обучения сети. Оказалось, что веса конкуренции во всем диапазоне

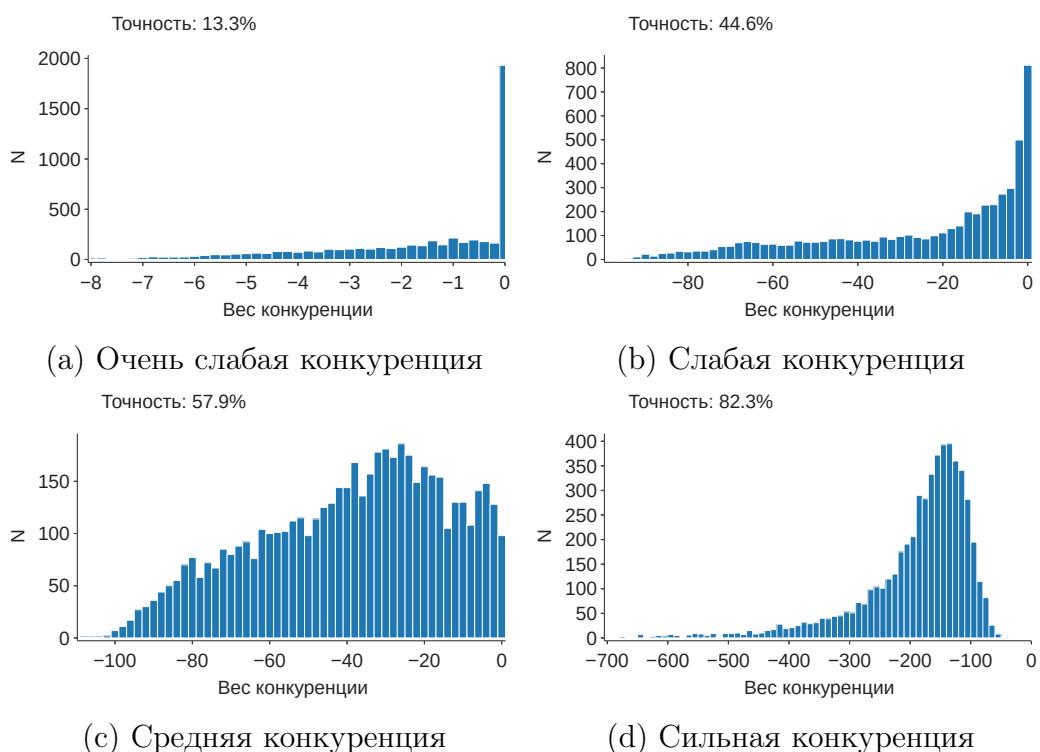
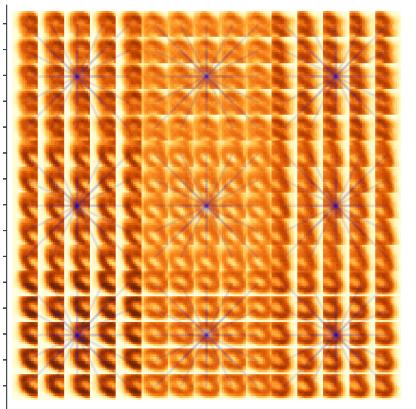
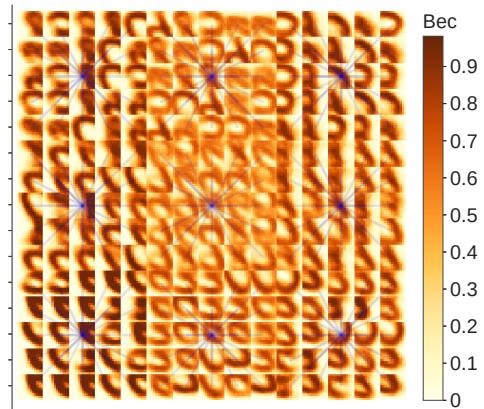


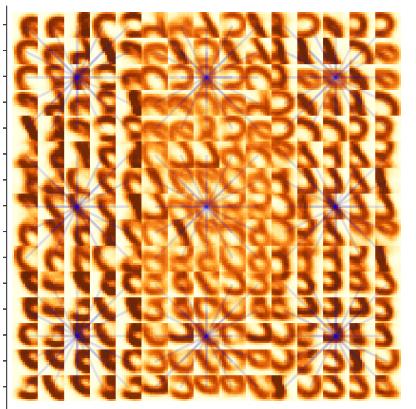
Рис. 6. Различные распределения весов конкуренции



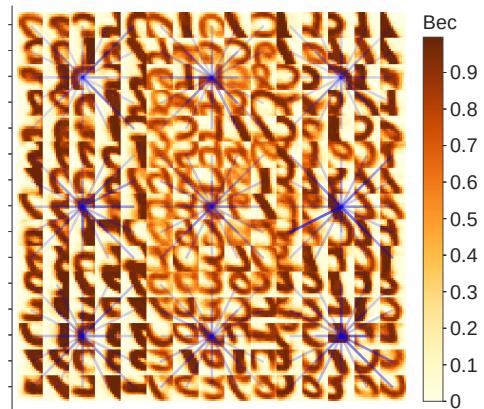
(a) Очень слабая конкуренция



(b) Слабая конкуренция



(c) Средняя конкуренция



(d) Сильная конкуренция

Рис. 7. Визуализация весов конкуренции поверх весов XY . Изображения соответствуют весам сетей с рисунка 6. Изображены только веса конкуренции для одного нейрона в каждом рецептивном поле для избегания загромождения визуализации. Насыщенный синий цвет соответствует большим по модулю весам конкуренции (используется среднее арифметическое между весами W_{ij} и W_{ji}). Видно, что похожие признаки сильнее конкурируют между собой, чем различные.

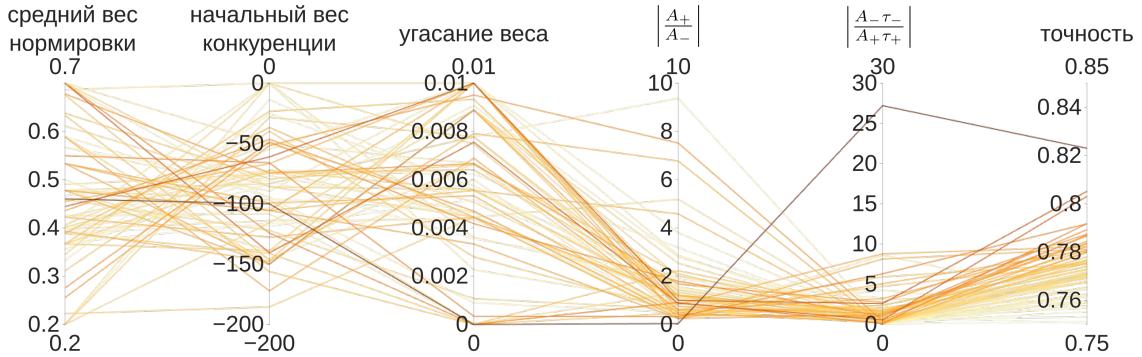


Рис. 8. Визуализация достижимой точности распознавания LCSNN при использовании различных комбинаций параметров в пространстве гиперпараметров. Параметры «вес нормировки», «начальный вес конкуренции» и «угасание веса» отвечают за значения весов, а следующие два являются комбинациями параметров anti-STDP. На последней оси указана точность сети с конфигурацией, соответствующих точкам пересечения ломаной осей гиперпараметров.

их изменения играют важную роль в работе LCSNN, поскольку ограничение как сверху, так и снизу негативно влияет на точность распознавания (Рис. 9). Это объясняется тем, что высокая конкуренция способствует большей специализации нейронов и потому полезна, а низкая конкуренция позволяет нейронам кооперироваться и распознавать классы совместно (в том числе обеспечивая накопление большей статистики для калибровки голосов нейронов).

Обучение конкуренции позволило достичь точности, незначительно (на 2%) превышающей точность сети такой же конфигурации, но с фиксированной конкуренцией с ингибирующими весами, равными -50 (Таб. 1, №4 и №5).

3 Обсуждение

Настоящее исследование демонстрирует, что локально соединенная сеть – перспективная спайковая нейросетевая архитектура, подхо-

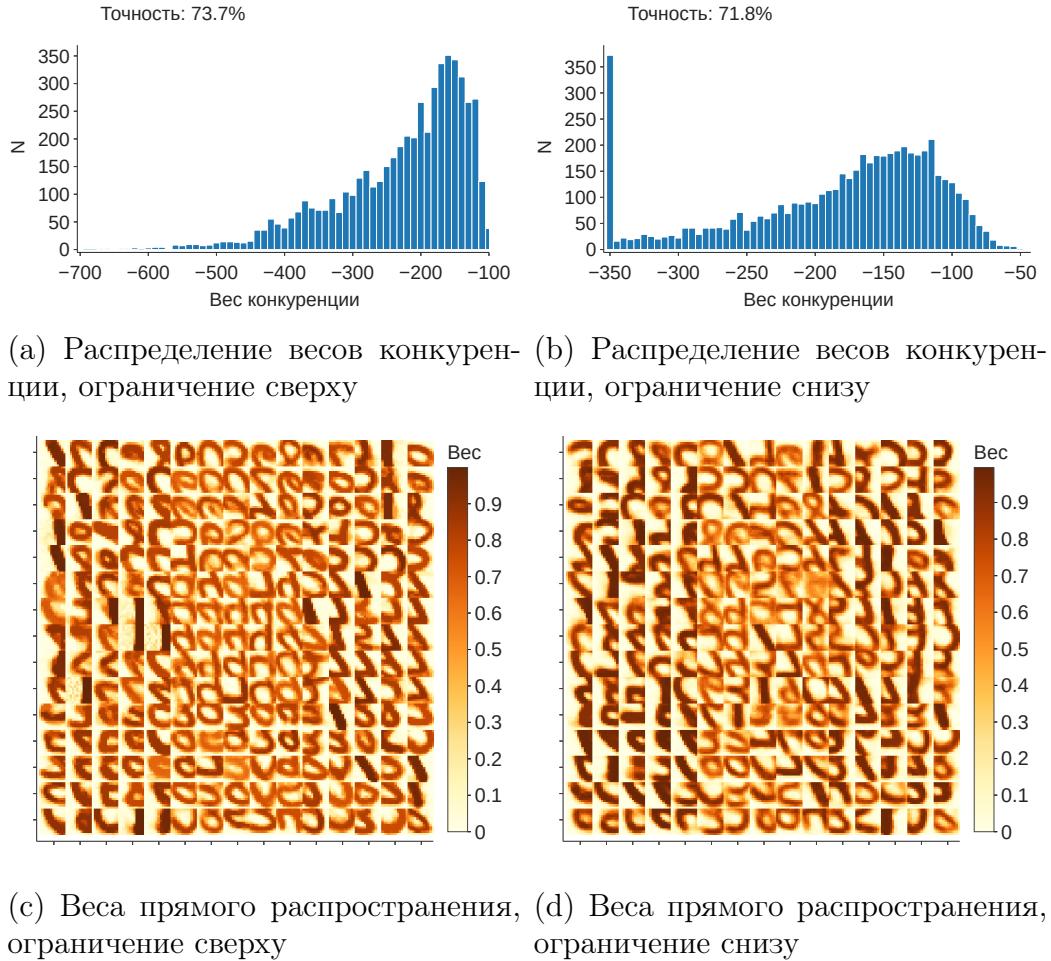


Рис. 9. Влияние ограничения значений весов конкуренции на точность. Веса были ограничены значениями -350 снизу и -100 сверху. Точность сети с аналогичными параметрами, но без ограничения конкуренции составляет 0.823 , ее веса XY изображены на Рис. 7d, а распределение ее весов конкуренции — на Рис. 6d. Видно, что в обоих случаях обучаются менее четкие веса прямого распространения.

дящая для эффективной реализации на специализированном нейрочипе. Действительно, способность быстро выходить на плато криевой обучения позволяет использовать LCSNN 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 сами признаки распознаваемых объектов выучиваются сетью без учителя. Остается открытым вопрос о возможности получения схожих результатов без использования обучения с учителем для интерпретации активности сети.

Главным преимуществом LCSNN является, как было показано, то, что локально соединенная сеть превосходит сверточную в точности при примерно одинаковом или чуть большем числе параметров. Предположительно, это происходит за счет более богатой статистики патчей (рецептивных полей), на которые реагируют нейроны внутри каждого канала, по сравнению с одним рецептивным полем на каждый канал в сверточной архитектуре. Ценой за это является увеличенное число параметров в LCSNN, но не является критическим фактором, ввиду локальной (в противоположность полносвязанной) топологии межнейронных контактов. Локальная архитектура также позволяет сохранить информацию о локализации выделяемого признака в пространстве и потому является разумным выбором для реализации SNN, обучающихся преимущественно без учителя. Данный выбор обусловливает также его совокупную эффективность (по производимости, точности и энергопотреблению при определённой плотности элементов) при аппаратной реализации, при которой каждый вес обычно представлен одним или несколькими физическими элементами (главным образом, ячейками SRAM [10; 11] или мемристорами [28]). Обучение связей конкуренции слегка увеличивает точность работы алгоритма на базе LCSNN, но не настолько критично, чтобы оправдывать использование такой дорогой операции как введение и обучение значительного числа ингибирующих связей с различными по модулю конечными весами. В то же время, вероятно, этот вывод справедлив только для небольших сетей, вроде исследованной в настоящей работе. На наш взгляд, более тща-

тельного изучения заслуживает подбор оптимальных алгоритмов и параметров обучения конкуренции для глубоких SNN, в которых баланс конкуренции и кооперации нейронов внутри каждого слоя может приводить к формированию кластеров из классов подаваемых изображений, что способствует выстраиванию семантически-подобной иерархии выученных признаков и их комбинаций, то есть конечных образов [29]. Потенциально такая иерархия образов может привести к существенному приросту качественных показателей работы интеллектуальных алгоритмов.

Выводы

Было показано, что для задачи распознавания образов на датасете MNIST локально соединенная архитектура с конкуренцией превосходит сверточную при равном числе параметров. Также было показано, что обучение весов конкуренции позволяет незначительно повысить качество модели, а в полученном распределении весов конкуренции важны как высокие по модулю (сильная конкуренция), так и низкие (коопeração) значения весов.

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. Language Models are Few-Shot Learners / T. B. Brown [и др.]. — 2020. — arXiv: [2005.14165 \[cs.CL\]](https://arxiv.org/abs/2005.14165).
6. 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>.

7. *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>.
8. *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).
9. *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>.
10. 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>.
11. 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).
12. Akida Neural Processor System-on-Chip. — 2020. — апр. — URL: <https://brainchipinc.com/akida-neuromorphic-system-on-chip/>.
13. 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).
14. 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.joa.2017.03.002](https://doi.org/10.1016/j.joa.2017.03.002).

j . sysarc . 2017 . 01 . 003 . — URL: <http://www.sciencedirect.com/science/article/pii/S1383762117300231>.

15. Tang G., Shah A., Michmizos K. P. Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM. — 2019. — arXiv: [1903.02504 \[cs.R0\]](https://arxiv.org/abs/1903.02504).
16. 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).
17. 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).
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 \[cs.NE\]](https://arxiv.org/abs/1904.06269).
22. Roberts P., Leen T. Anti-Hebbian Spike-Timing-Dependent Plasticity and Adaptive Sensory Processing // Frontiers in computational neuroscience. — 2010. — дек. — т. 4. — с. 156. — DOI: [10.3389/fncom.2010.00156](https://doi.org/10.3389/fncom.2010.00156).
23. Hadsell R., Chopra S., LeCun Y. Dimensionality Reduction by Learning an Invariant Mapping. — 2006. — DOI: [10.1109/CVPR.2006.100](https://doi.org/10.1109/CVPR.2006.100).

24. *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>.
25. *Tavanaei A., Maida A. S.* Multi-layer unsupervised learning in a spiking convolutional neural network. — 2017.
26. 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>.
27. *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.
28. 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%2F1361-6463%2Faade3f>.
29. *Nekhaev D., Demin V.* Competitive Maximization of Neuronal Activity in Convolutional Recurrent Spiking Neural Networks. — 2020.

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