

# Методы Обучения Импульсных Нейронных Сетей

Андрей В. Гаврилов<sup>1</sup>, Константин О. Панченко<sup>2</sup>

<sup>1</sup>Новосибирский государственный технический университет, Новосибирск, Россия

<sup>2</sup>ООО «Мотив», Новосибирск, Россия

**Аннотация** – В докладе дан обзор методов обучения импульсных нейронных сетей по зарубежным источникам. Рассмотрены следующие виды обучения: метод STDP (модифицированное правило Хебба), обучение обратным распространением ошибки, метод ReSuMe, методы линейной алгебры, растущие импульсные нейронные сети, глубокое обучение (deep learning).

**Ключевые слова** – Импульсные нейронные сети, машинное обучение, нейроморфные технологии, алгоритмы обучения нейронных сетей.

## I. ВВЕДЕНИЕ

ПОСЛЕДНЕЕ десятилетие ознаменовалось бурным развитием третьего поколения нейронных сетей – импульсных нейронных сетей (spiking neural networks SNN). Это было вызвано появлением возможности разрабатывать и производить большие аппаратные нейронные сети, реализованные в виде чипов, что позволило говорить о возможности замены во многих областях применения традиционной фон Неймановской архитектуры компьютеров на нейросетевую, основанную на импульсных нейронных сетях с низким энергопотреблением [1]. Эта замена требует решения следующих вопросов:

- какие методы обучения должны быть основными,
- как обучать SNN: на хост-компьютере предварительно в режиме off-line или в процессе эксплуатации нейрочипа в реальном времени,
- если в реальном времени, то как может быть реализовано обучение нейронной сети в чипе, построенном по технологии CMOS, FPGA или других технологий,
- как обеспечить масштабируемость и интеграцию нейронных сетей при построении суперкомпьютеров для обработки Big Data или для моделирования функций мозга,
- как интегрировать нейросетевые чипы с сенсорами и актуаторами,
- как кодировать входные и выходные данные для импульсной нейронной сети.

В публикациях, обзор которых предлагается в этой статье, внимание уделяется главным образом первым трем вопросам.

Возможны два подхода к выбору методов обучения импульсных нейронных сетей:

- попытаться перенести известные методы обучения классических нейронных сетей на импульсные нейронные сети (например, метод обучения обратным распространением ошибки),
- создавать новые методы и алгоритмы с учетом особенностей импульсных нейронных сетей.

Можно выделить следующие основные виды методов обучения импульсных нейронных сетей, которые рассматриваются далее в этой статье:

- метод STDP (модифицированное правило Хебба),
- обучение обратным распространением ошибки,

- управляемый метод обучения Хебба (supervised Hebbian learning),
- метод ReSuMe,
- растущие импульсные нейронные сети,
- глубокое обучение (deep learning).

## II. МЕТОД STDP (SPIKE TIMING DEPENDENT PLASTICITY)

Как известно, предложенное Хеббом в 1940х годах правило обучения основано на нейрофизиологических данных и гласит применительно к классическим нейронным сетям, что связи между нейронами с похожими состояниями усиливаются, а между разными состояниями ослабевают. В модификации этого правила для импульсных нейронных сетей, в методе STDP (spike timing dependent plasticity) изменение весов синапсов нейрона зависит от разницы во времени между пре-синаптическим и пост-синаптическим спайком, т.е. от предполагаемой корреляции между пришедшим на синапс нейрона импульсом и сгенерированным этим же нейроном выходным импульсом.

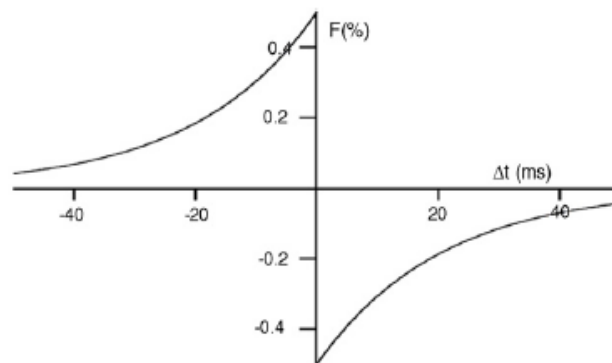


Рис. 1. Зависимость величины изменения веса синапса от разницы во времени между пост-синаптическим и пре-синаптическим спайком.

В [4] описан пример применения метода STDP для обучения импульсной нейронной сети распознаванию черно-белых (с оттенками серого) изображений десятичных цифр из базы данных MNIST. Величина интенсивности каждого из  $28 \times 28$  пикселей преобразуется в случайную последовательность импульсов, частота (вероятность) появления которых в каждом такте пропорциональна интенсивности пиксела. Последовательность от каждого пиксела является входной информацией для каждого из множества возбуждающих нейронов. Каждый из возбуждающих нейронов связан по выходу с одним тормозящим нейроном. Каждый из тормозящих нейронов связан по выходу со всеми возбуждающими нейронами кроме того, который ему соответствует. Таким образом, тормозящие нейроны реализуют латеральные связи между возбуждающими нейронами. Распознавае-

мые классы в процесс обучения не предъявляются сети, т.е. используется обучение без учителя. Возбуждающие нейроны связываются с классами после обучения, которое основано на выявлении максимально в среднем активного нейрона, соответствующего определенному классу. Никакие дополнительные параметры не используются для предсказания класса, в частности, какие-либо линейные классификаторы или подобные методы на выходе нейронной сети.

Кроме синаптического веса каждый синапс хранит параметр  $x_{pre}$ , моделирующий пре-синаптическую историю прихода импульсов. В каждом такте, если на синапс приходит импульс, то  $x_{pre}$  увеличивается на 1, в противном случае экспоненциально уменьшается. Когда срабатывает нейрон, изменение веса его каждого синапса  $\Delta w$  вычисляется на основе истории активности синапса как:

$$\Delta w = \eta(x_{pre} - x_{tar})(w_{max} - w)^{\mu} \quad (1)$$

где:

$\eta$  - коэффициент, задающий скорость изменения веса синапса,

$w_{max}$  - максимальная величина веса,

$\mu$  задает зависимость изменения веса от предыдущего веса,

$x_{tar}$  - целевое значение параметра  $x_{pre}$  при срабатывании нейрона.

Формула (1) гарантирует, что тот синапс, который редко приводит к срабатыванию нейрона, будет все более и более снижать свой вес.

Распределение возбуждающих нейронов по классам (присваивание распознаваемым классам меток) проводится после обучения путем выявления наиболее активного нейрона при предъявлении нейронной сети типичного представителя соответствующего класса.

В [5] метод STDP применялся для управления мобильным роботом (виртуальным насекомым) для решения задачи движения к цели среди обнаруживаемых в процессе движения препятствий. На рис. 2 показана структура нейронной сети, использованной для решения этой задачи. Четыре входных нейрона 1, 2, 4 и 5 воспринимают информацию от препятствия слева и справа и цели слева и справа, соответственно. Два выходных (моторных) нейрона обеспечивают управление скоростью двух колес. Авторы использовали радиально-базисную модель импульсного нейрона [6].

Моделирование проводилось в MATLAB с созданием виртуального пространства 600x600 пикселей. При моделировании оценивались три состояния сети (исходная, частично обученная и полностью обученная) в трех вариантах виртуального пространства: пустое (только цель без препятствий), лабиринт (с препятствиями) и облако (с частыми препятствиями и узкими проходами между ними). Моделирование показало применимость обучения методом STDP для управления мобильным роботом.

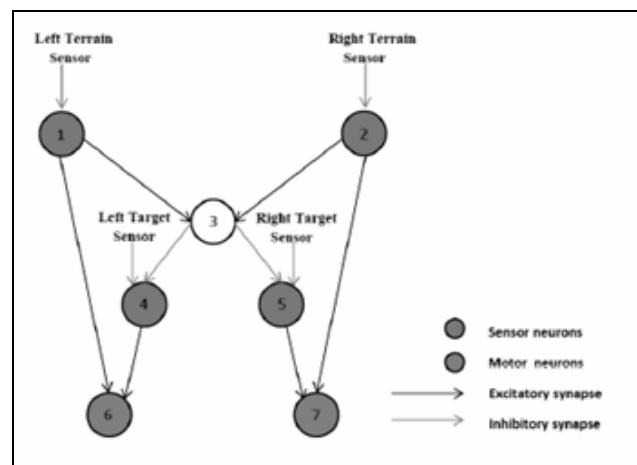


Рис. 2. Импульсная нейронная сеть для управления мобильным роботом.

Недостатком применения модифицированного правила Хебба в виде метода STDP является то, что он обеспечивает только обучение без учителя, хотя известны попытки комбинирования STDP с обучением обратным распространением ошибки.

### III. ОБУЧЕНИЕ ОБРАТНЫМ РАСПРОСТРАНЕНИЕМ ОШИБКИ

Основной проблемой при применении для импульсных нейронных сетей обучения с учителем, в частности, методом обратного распространения ошибки, является бинарный характер входов и выходов импульсных нейронов. Поэтому, большинство градиентных методов обучения не может быть использованы непосредственно для импульсных нейронных сетей.

Тем не менее, в начале 2000-х годов был предложен алгоритм обучения импульсной нейронной сети SpikeProp [7], основанный на обратном распространении ошибки. Авторы использовали в качестве непрерывных переменных времена появления спайков на входах и выходах нейронов, а в качестве ошибки использовали разницу между ожидаемым и реальным временем появления импульса. Таким образом, в этом подходе импульс интерпретируется как событие (event) и ни о каком модулировании сигнала речи не идет.

Алгоритм SpikeProp был изучался, модифицировался и улучшался в [8, 9, 10, 11]. Было обнаружено, что инициализация весов является критическим фактором для получения хороших результатов обучения. В [8] было доказано, что обучение с высоким коэффициентом обучения (задающим скорость изменения весов) и с отрицательными весами не может обеспечить успешную сходимость процесса обучения. Эти результаты противоречили выводам, сделанными в [10], где авторы предложили модификацию алгоритма обучения, а именно, введение расчета момента в уравнение изменения весов. Было продемонстрировано, что эта модификация значительно ускоряет сходимость алгоритма SpikeProp. В [9] авторы предложили в градиентный алгоритм обучения SpikeProp ввести помимо изменения весов синапсов еще задержки сигнала в синапсах, константу времени и пороги срабатывания нейронов. Это приводит к ускорению сходимости обучения и к упрощению топологии нейронной сети, необходимой для решения определенной задачи. В [11] авторы развили алгоритм SpikeProp до применения в рекуррентных импульсных нейронных сетях, где обеспечивается учет временных зависимостей в потоке

входных данных. Но ни оригинальный алгоритм SpikeProp ни его модификации, описанные выше, не имеют дело с паттернами, состоящими из последовательности импульсов.

Алгоритм SpikeProp был протестирован на классической задаче обучения распознаванию ирисов. Были получены результаты не хуже, чем при использовании в Matlab алгоритмов BP (back propagation) и LM (Левенберга – Марквардта).

Недостаток кодирования информации в методе SpikeProp это некоторые трудности при интерпретации результатов распознавания и преобразовании временного интервала в более удобную для использования форму, например, для управления мобильным роботом. Кроме того, необходима специфическая предварительная обработка сенсорной информации для подачи ее на нейронную сеть. Предположение о кодировании одним входным импульсом некоторого события сужает область применимости алгоритма SpikeProp и ему подобных алгоритмов. Более того, главным препятствием для реального применения алгоритма SpikeProp является то, что он не обеспечивает какой-либо механизм изменения весов в случае, когда нейрон перестает генерировать спайк в ответ на обучающий пример.

В статье [12] дано хорошее сравнение методов обучения с учителем при использовании кодирования в виде времени появления спайка (time-spiking encoding).

Другой подход, основанный на использовании обратного распространения ошибки, был предложен недавно для обучения нейро-чипа TrueNorth компании IBM [13]. В этом случае авторы используют оффлайн обучение на хост-компьютере и предлагают для этого заменить входные и выходные импульсы вероятностями их появления, что позволяет применить градиентный метод обучения.

Таким образом, обучение в соответствии с методом обратного распространения ошибки выполняется итерационно в виде следующих шагов:

- 1) прохождение входных сигналов через сеть от входного слоя до выходного слоя,
- 2) сравнение выходов сети с желаемыми выходами и вычисление функции ошибки,
- 3) обратное распространение ошибки с вычислением для каждого синапса промежуточных слоев градиента функции ошибки,
- 4) использование полученного градиента для изменения параметров синапсов.

Обучаемая нейронная сеть является вероятностным представлением импульсной нейронной сети. Синаптические связи представляются как вероятности  $\zeta_{ij}$ , такие, что  $P(c_{ij}=1) = \zeta_{ij}$ , синаптический вес представляется как  $s_{ij}$  так же, как и в сети нейро-чипа. Предполагается, что переменные  $s_{ij}$  определены на ограниченном диапазоне значений и на них действует дополнительное ограничение, которое заключается в том, что существуют «блоки», в которых синапсы имеют одинаковые значения весов, как это сделано в TrueNorth.

После обучения нейронная сеть с полученными в процессе обучения значениями вероятностей синаптических связей в ядре TrueNorth преобразуется в импульсную сеть для размещения в чипе. При этом для каждого распознаваемого класса создается ансамбль из нескольких нейронных сетей, в которых случайно формируется матрица синаптических связей  $c_{ij} = (0, 1)$  в ядре так, чтобы значения 0 и 1 в этой матрице по всем сетям из ансамбля соответствовали полученным в

процессе обучения вероятностям связей. Эта матрица задает наличие связей между  $i$ -м синапсом и  $j$ -м нейроном. Суммарный сигнал на  $j$ -м нейроне вычисляется по формуле:

$$I_j = \sum_i x_i c_{ij} s_{ij} + b_j \quad (2)$$

где:

$x_i$  – входной сигнал на  $i$ -м синапсе,

$s_{ij}$  – вес (сила)  $i$ -го синапса применительно к  $j$ -му нейрону,

$b_j$  – смещение  $j$ -го нейрона.

Следует заметить, что авторы не изменяют веса в процессе обучения, а изменяют только вероятности синаптических связей. Веса  $s_{ij}$  задаются в процессе решения оптимизационной задачи минимизации избыточности нейронов.

Этот подход был протестирован при решении хорошо известной задачи классификации ирисов. При этом были получены хорошие результаты для разного количества нейронных сетей в ансамбле: 1, 4, 16 и 64. Получены вероятности правильного распознавания от 92.7% до 99.42%.

#### IV. УПРАВЛЯЕМЫЙ МЕТОД ОБУЧЕНИЯ ХЕББА (SUPERVISED HEBBIAN LEARNING)

В [28] был предложен управляемый метод Хебба (Supervised Hebbian Learning, SHL), объявленный как наиболее близкий к нейрофизиологическим данным метод обучения импульсных нейронных сетей. Метод SHL направлен на то, чтобы выходной сигнал появлялся в желаемое время. Для этого в нем используется так называемый «обучающий» сигнал. Since the teaching signal comprises of intracellular synaptic currents, supervision may be envisioned as supervision by other neurons. Метод SHL имеет недостаток, который заключается в том, что даже после того, как желаемый паттерн на выходе достигнут, алгоритм обучения продолжает изменять веса. Таким образом, должны быть введены ограничения для обеспечения стабильности. Однако, проблема с установкой ограничений не проста, т.к. трудно узнать, когда в процессе обучения они должны быть применены. Веса изменяются после каждой эпохи, что может привести к нестабильности или появлению при тестировании неудовлетворительных результатов при тестировании [12]. Авторы в [28] предложили одну из первых реализаций подхода SHL. Они определили правило обучения для нейрона с одним возбуждающим синапсом. Процесс обучения был основан на трех спайках (два пре-синаптический и один пост-синаптический), генерируемых в каждом цикле обучения. Первый пре-синаптический импульс в момент времени  $t_1^{in}$  воспринимался как входной сигнал, второй пре-синаптический импульс в момент времени  $t_2^{in}=t^d$  указывал на желаемое время срабатывания пост-синаптического нейрона. Обучающее правило имело следующий вид:

$$\Delta w = \eta(t^{out} - t^d),$$

где  $\eta > 0$  коэффициент скорости обучения и  $t^{out}$  реальное время появления пост-синаптического спайка.

В [29] авторы предложили модифицированный метод SHL, основанный на концепции нечеткой импульсной нейронной сети (Fuzzy Spiking Neural Network FSNN) и кодирования информации в виде частотной модуляции.

Входной нейрон обеспечивает кодирование значения первичного признака в виде соответствующей частоты импульсов. Последовательность импульсов генерируется по схеме

линейного кодирования. Схема кодирования преобразует значение частоты в между-спайковый интервал (inter-spike-interval ISI), который используется для создания последовательности входных импульсов. Каждый элемент входных данных преобразуется в последовательность импульсов определенной длины.

Гауссова радиальная функция (RF) размещается в каждом синапсе между входным и промежуточным нейронами. Назначение RF определять отношение между входными частотами  $f_i$  и центральной частотой  $F_0$  функции RF. Вес определяется как  $k_i$ . Процесс имеет отношение к части нечеткого правила IF ( $x_i$  is  $A_i$ ), где  $x_i$  вход и  $A_i$  представляет RF. Т.к. используется радиальная функция RF, нет необходимости точно настраивать динамический синапс. Назначение каждого скрытого нейрона представлять оставшуюся часть нечеткого правила – его antecedent, конъюнкцию, суммирование пост-синаптического потенциала (PSP) представляется как дизъюнкция. Основное назначение функции RF скрытых нейронов – фильтрация спайков на пути к выходному слою.

Выходном слое потенциал действия синапса имеет высокое значение только очень короткое время. Такой тип синапсов можно назвать детектором совпадений. Задачей обучающего алгоритма является создание ассоциаций между скрытыми и выходными нейронами, таким т.е. нечетких отношений между antecedentом и консеквентом.

Авторы использовали метод STDP с обучающими окнами для модификации силы связей между скрытыми и выходными нейронами. Эта модифицированная форма подхода SHL не требует обучающего спайка как в [28].

Нечеткая импульсная нейронная сеть была реализована в среде MATLAB.

## V. МЕТОД ReSuMe

В [14, 15] был предложен метод обучения RuSuMe, основанный на так называемом «удаленном обучении» (remote supervision), использующий множественные спайки. В нем нейронная сеть учится генерировать заданную последовательность импульсов в ответ на входную последовательность.

Метод ReSuMe объединяет достоинства метода Хебба с концепцией удаленного обучения. Понятие «удаленное обучение» означает, что желаемые сигналы не направляются непосредственно на обучаемые нейроны (как это происходит в методе SHL). Они участвуют в изменении синаптических весов связей, заканчивающихся на обучаемых нейронах. А именно, синаптический вес  $w_{ki}$ , между некоторым пре-синаптическим нейроном  $n_k^{in}(i)$  и соответствующим пост-синаптическим нейроном  $n_i^{out}$ , модифицируется в соответствии с двумя правилами. Первое правило зависит от корреляции между временами срабатывания нейронов  $n_k^{in}(i)$  и  $n_i^{out}$ . Второе правило определяется корреляцией между временами срабатывания нейронов  $n_k^{in}(i)$  и  $n_j^d(i)$ . Под  $n_j^d(i)$  подразумевается обучающий нейрон, вырабатывающий желаемый сигнал для  $n_i^{out}$ . Для возбуждающих синапсов эти два правила соответствуют правилам увеличения и уменьшения весов в методе STDP и описываются с помощью окон обучения (learning windows)  $W^d(s^d)$  и  $W^{out}(s^{out})$  (Рис. 3).

Метод ReSuMe был проверен при обучении SNN восстановлению пространственно-временных паттернов из спайков, соответствующих активности пула моторных нейронов. Каждый пул, состоящий из 40 нейронов, активировал свою модель мускула. Упрощенная модель конечности, управляе-

мая импульсной нейронной сетью, была способна следовать по желаемой траектории с удовлетворительной точностью.

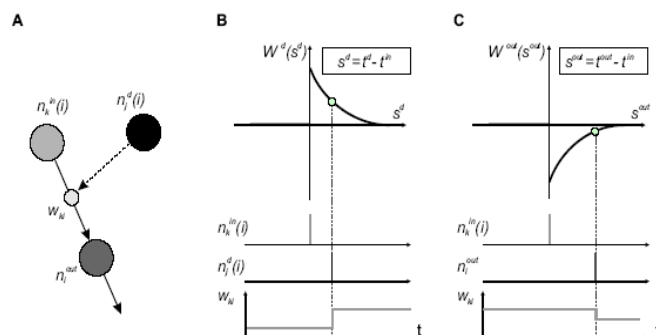


Рис. 3. Концепция удаленного обучения и окон обучения в ReSuMe.

## VI. РАСТУЩИЕ ИМПУЛЬСНЫЕ НЕЙРОННЫЕ СЕТИ

Некоторые исследователи используют для обучения импульсных нейронных сетей эволюционное программирование, основанное на использовании генетических алгоритмов. Этот подход был предложен в [18] для обучения SNN с ограничениями на величину кода веса и задержки синапса тремя битами. Авторы использовали кодирование выходных данных сети в виде времени появления выходного импульса и среднеквадратическое отклонение для этой величины в качестве фитнес-функции. Они протестировали этот подход на задаче классификации ирисов. При этом были получены достаточно хорошие результаты, и было проведено их сравнение с аналогичными результатами, полученными при использовании алгоритма SpikeProp и его модификаций. Однако, в [18] не использовалась возможность применения генетических алгоритмов для создания растущих импульсных нейронных сетей (с созданием новых нейронов и связей).

В [19] предложен метод обучения, основанный на применении генетических алгоритмов, для генерации и тестирования растущих импульсных сетей. Геном в этом подходе содержит не только параметры синапсов и нейронов, но и топологию сети. В экспериментах обученная импульсная нейронная сеть демонстрировала генерирование выходного паттерна импульсов (например, для управления походкой мобильного робота) в ответ на входную последовательность и устойчивость к помехам во входном паттерне. Другие задачи, которые решались с помощью этого подхода: обнаружение заданной входной последовательности, обнаружение искажения входного паттерна.

Топология нейронной сети описывалась матрицей связей, которая кодировалась с помощью специального алгоритма для размещения в хромосоме. Кроме топологии в хромосоме размещались следующие параметры для каждого нейрона: рефрактерный период, временная константа задающая крутизну экспоненты возврата мембранного потенциала к исходной величине. Кроме того, в хромосоме кодировались следующие параметры каждого синапса: вес, задержка, временные константы, определяющие экспоненциальный рост и уменьшение пост-синаптического потенциала.

Подобный подход, основанный на применении генетических алгоритмов, был предложен в [20]. Здесь используется меньшее количество параметров нейронов для обучения сети и решается задача классификации. Предложенные авторами алгоритмы были протестированы на задачах классификации ирисов, диагностики рака груди и гепатита.

Недостатком применения генетических алгоритмов для обучения импульсных нейронных сетей является то, что для обучения требуется много времени. Достоинством является то, что возможно для обучения можно использовать практически любые параметры нейронов, синапсов и сети в целом.

В статье [21] авторы предложили другой подход для построения растущих импульсных нейронных сетей без применения генетических алгоритмов. Авторы предложили новый класс импульсных нейронных сетей – dynamic evolving spiking neural networks (deSNN) или динамические развивающиеся импульсные нейронные сети. Эти сети ориентированы на on-line обучение в реальном времени и на распознавание пространственно-временных паттернов. Этот подход основан на концепции развивающихся импульсных нейронных сетей, предложенной ранее в [21] и в какой-то степени похож на теорию адаптивного резонанса для классических нейронных сетей [22]. В этом подходе авторы используют также Евклидово расстояние между входным вектором и центром кластера в алгоритме, решающим создавать новый нейрон (новый кластер) или нет. В последнем случае входной паттерн распознается.

Авторы в deSNN используют кодирование информации в виде порядка появления импульсов (rank-order learning) и алгоритм SDSP (Spike Driven Synaptic Plasticity) – вариант STDP. При этом они использовали три варианта режимов обучения: без учителя (unsupervised), с учителем (supervised) и semi-supervised. Алгоритм SDSP использовался для динамического изменения весов связей для обнаружения пространственно-временных паттернов как при обучении так и при использовании обученной сети. Предложенная модель deSNN была проверена при решении двух задач: 1) распознавание движущегося объекта *verified on two case study applications*: (1) moving object recognition и 2) распознавание электроэнцефалограммы в системе взаимодействия «мозг-компьютер».

## VII. ГЛУБОКОЕ ОБУЧЕНИЕ

Нейронные сети глубокого обучения (deep learning), такие как сверточные сети (convolutional neural networks, CNN), в последнее десятилетие получили большое распространение для решения сложных задач технического зрения, например, распознавания объектов.

В [24, 25, 26] предлагается подход, основанный на конвертировании сверточной сети в импульсную нейронную сеть.

В статье [24] предлагается метод конвертирования CNN в импульсную нейронную сеть, реализованную в чипе. Сначала создается сверточная сеть с учетом требований импульсной сети, в которую она будет конвертирована. Затем, сверточная сеть обучается как обычно, и обученная сеть с полученными весами связей преобразуется в импульсную для размещения в чипе. Полученная импульсная нейронная сеть была протестирована на базах данных Neovision2 Tower и CIFAR-10, и показала похожие результаты по точности распознавания, что и исходная сверточная сеть. Полученная при этом импульсная сеть может быть реализована в чипе, подобном TrueNorth, разработанному компанией IBM. Авторы показали, что такая сеть, размещенная в чипе, на два порядка менее энергоемкая, чем обычная сверточная сеть, реализованная в технологии FPGA (ПЛИС).

В статье [25] авторы обучали импульсную глубокую нейронную сеть, использующую LIF-нейроны (leaky integrate-and-fire) и получили хорошие результаты при работе с

базами данных CIFAR-10 и MNIST. Они достигли этих результатов путем «размытия» активационной функции LIF-нейрона так, что ее производная имеет ограниченное значение. Кроме того, они при обучении использовали шум для обеспечения устойчивости.

После обучения обученная «статическая» сеть конвертируется в динамическую импульсную нейронную сеть. Параметры импульсной сети (веса и смещения) идентичны обученной «статической» сети. Операция свертки остается такой же, хотя она может быть переписана как матрица весов  $w_{ij}$  связей между пре-синаптическим нейроном  $i$  и пост-синаптическим нейроном  $j$ . Таким же образом, операция объединения (average pooling operation) может быть переписана как простая матрица весов связей, и эта матрица может быть помножена на матрицу свертки следующего слоя для того, чтобы получить веса связей между нейронами.

Этот подход может быть применим для других типов нейронов, в том числе, используемых в современных аппаратно-реализованных нейронных сетях.

В [26] авторы анализируют особенности конвертирования обычных сетей в импульсные нейронные сети с учетом выбора параметров импульсной нейронной сети, таких как скорость срабатывания и порог нейрона. Авторы предлагают множество техник оптимизации для минимизации потерь при конвертировании сверточных и полносвязных глубоких нейронных сетей. Применение этих техник позволяет получить импульсные нейронные сети, превосходящие все предыдущие при тестировании на базе данных MNIST, и сети, которые показывают хорошие результаты уже после 20 миллисекунд времени моделирования. Эти техники включают в себя использование так называемых «сглаженных линейных блоков» (rectified linear units - ReLU) с нулевым смещением в процессе обучения, метод нормализации новых весов для регулирования скорости срабатывания нейрона.

В [27] авторы разработали схему обучения связями в глубокой нейронной сети. Эта схема основана на использовании условных вероятностей для управления частотой срабатывания нейрона, вычисление которой основано на множестве статистических методов, разработанных в рамках исследований в области глубокого обучения. Этот подход был протестирован на базе данных MNIST.

## VIII. ЗАКЛЮЧЕНИЕ

Как видно из данного краткого обзора, существует множество различных методов и алгоритмов обучения импульсных нейронных сетей. Наиболее используемыми из них являются модифицированный метод Хебба STDP (Spike Timing Dependent Plasticity) и его модификации, а также, метод обратного распространения ошибки SpikeProp и его модификации (для кодирования информации временем появления одиночного импульса). Другие методы кодирования информации почти не используются, в частности, частотная модуляция. В последнее время в связи с появлением нейро-чипов, подобных TrueNorth компании IBM, появились подходы, основанные на обучении и генерации нейронных сетей в офф-лайн режиме (в том числе, сверточных сетей) и конвертирования обученной нейронной сети в импульсную нейронную сеть, реализуемую в чипе.

В результате проведенного аналитического обзора можно сформулировать актуальные проблемы обучения импульс-

ных нейронных сетей, которые необходимо решить для их широкого применения:

- разработка формализованных, эффективных и универсальных методов конвертирования обученной классической нейронной сети в импульсную нейронную сеть для размещения ее в чипе,
- разработка растущих импульсных нейронных сетей без использования генетических алгоритмов, подобных классическим сетям адаптивного резонанса,
- разработка более био-подобных, чем SpikeProp, алгоритмов обучения с учителем,
- исследования возможностей построения обучаемых в реальном времени импульсных нейронных сетей (в том числе, растущих), реализованных в чипе.

### СПИСОК ЛИТЕРАТУРЫ

- [1] Andrew S. Cassidy a,1, Julius Georgiou b, Andreas G. Andreou. Design of silicon brains in the nano-CMOS era: Spiking neurons, learning synapses and neural architecture optimization // *Neural Networks*, 2013, № 45, pp. 4–26.
- [2] Bi,G.-Q., Poo,M. (1998). Synaptic modifications in cultured hippocampal neurons :dependence on spike timing, synaptic strength, and postsynaptic cell type // *J.Neurosci.*, 1998 18, pp. 10464–10472.
- [3] Song S, Miller KD, Abbott LF. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity // *Nat Neurosci.*, 2000, 3(9), .pp. 919-926.
- [4] Peter U.Diehl, Matthew Cook..Unsupervised learning of digit recognition using spike-timing-dependentplasticity // *Frontiers in Computational Neuroscience*, August 2015, Volume 9, Article 99.
- [5] Zhang X., Xu \_Z., Henriquez C., Ferrari S. Spike-Based Indirect Training of a Spiking Neural Network-Controlled Virtual Insect // *Proc. of 52nd IEEE Conference on Decision and Control* December 10-13, 2013. Florence, Italy.
- [6] Zhang X., Foderaro G., Henriquez C., VanDongen A. M. J., Ferrari S., A radial basis function spike model for indirect learning via integrate-and-fire sampling and reconstruction techniques // *Advances in Artificial Neural Systems*, Volume 2012 (2012), Article ID 713581.-<http://dx.doi.org/10.1155/2012/713581>
- [7] Sander M. Bohte; Joost N. Koko, Han La Poutr. Error-backpropagation in temporally encoded networks of spiking neurons // *Neurocomputing*, 48, 2002, pp. 17–37.
- [8] Moore S. C.. Back-Propagation in Spiking Neural Net-works. MSc thesis, University of Bath, 2002.
- [9] Schrauwen B., Campenhout J. V. Improving SpikeProp: Enhancements to an Error-Backpropagation Rule for Spiking Neural Networks // *Proceedings of the 15th ProRISC Workshop*, 11 2004.
- [10] Xin J., Embrechts M. J. Supervised Learning with Spiking Neuron Networks // *Proceedings IEEE International Joint Conference on Neural Networks, IJCNN'01*, Washington D.C., July 15-19 2001, pp. 1772–1777.
- [11] Tiño P. Mills A. J. Learning Beyond Finite Memory in Recurrent Networks of Spiking Neurons. In L. Wang, K. Chen, and Y. Ong, editors, *Advances in Natural Computation // ICNC 2005*, Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 666–675.
- [12] Kasinski A., Ponulak F. Comparison of Supervised Learning Methods for Spike Time Coding in Spiking Neural Networks // *Int. J. Appl. Math. Comput. Sci.*, 2006, Vol. 16, No. 1, pp. 101–113.
- [13] Esser S.K, Appuswamy R., Merolla P.A., Arthur J.V., Modha D.S. Back-propagation for Energy-Efficient Neuromorphic Computing // *Proc. of Int. Conf. Neural Information Processing Systems 28 (NIPS 2015)*, Montreal, Canada, 2015.
- [14] Ponulak F., Kasiński A., Supervised learning in spiking neural networks with ReSuMe sequence learning, classification, and spike shifting // *Neural Comput.*, 2010, vol. 22, pp. 467-510.
- [15] Ponulak F.. Supervised learning in Spiking Neural Networks with ReSuMe Method. Doctoral Dissertation. Poznań University of Technology, 2006.
- [16] Carnell A., Richardson D. Linear Algebra for Time Series of Spikes, *Proc. of ESANN'2005 - European Symposium on Artificial Neural Networks Bruges (Belgium)*, 27-29 April 2005, pp. 363-368.
- [17] Cohen H.. *A Course in Computational Algebraic Number Theory*. - Springer-Verlag, New York, 1993.
- [18] Belatreche A., Maguire L. P., McGinnity M., Wu Q. X. A Method for Supervised Training of Spiking Neural Networks. // *Proceedings of IEEE Cybernetics Intelligence - Challenges and Advances (CICA)*, 2003, Reading, UK, pp. 39–44.
- [19] Schaffer J. D.. Evolving spiking neural networks: A novel growth algorithm corrects the teacher // *Proc. of 2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*. 26-28 May 2015, pp. 1-8.
- [20] Abdulrazak Yahya Saleh, Haza Nuzly Bin Abdull Hameed, Mohd Najib Mohd Salleh. A Novel hybrid algorithm of differential evolution with evolving spiking neural network for pre-synaptic neurons optimization // *Int. J. Advance Soft Compu. Appl*, Vol. 6, No. 1, March 2014.
- [21] Kasabov N., Dhole K., Nuntalid N., Indiveri G. Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition // *Neural Networks*, 41, 2013, pp. 188–201.
- [22] Wysoski, S., Benuskova L., Kasabov N. On-line learning with structural adaptation in a network of spiking neurons for visual pattern recognition // *Artificial Neural Networks–ICANN 2006*, 2006, pp. 61-70.
- [23] Carpenter G., A., Grossberg S. *Pattern Recognition by Self-Organizing Neural Networks*. - Cambridge, MA, MIT Press, 1991.
- [24] Yongqiang Cao, Yang Chen, Deepak Khosla. Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition // *Proc. of International Journal of Computer Vision*. May 2015, Volume 113, Issue 1, pp 54-66.
- [25] Hunsberger E., Eliasmith C. Spiking Deep Networks with LIF Neurons. - arXiv:1510.08829, 2015.
- [26] Diehl P.U. Neil D., Binas, J., Cook, M., Liu, S.C., Pfeiffer, M. Fast-Classifying, High-Accuracy Spiking Deep Networks Through Weight and Threshold Balancing // *Proc. of IEEE International Joint Conference on Neural Networks (IJCNN)*, 2015.
- [27] J. A. Henderson, T. A. Gibson, J. Wiles. Spike Event Based Learning in Neural Networks. - arXiv:1502.05777, 2015.
- [28] Ruf, B., Schmitt, M. Learning temporally encoded patterns in networks of spiking neurons // *Neural Processing Letters*, 5(1), 1997, pp. 9-18.
- [29] Rasa H., Tsegaye, Biswas R. Fuzzy based modified SHL algorithm for spiking neural networks // *Int. Journal of Computer Applications*, vol. 41, N. 5, March, 2012, pp. 33-37.



Андрей В.Гаврилов окончил Новосибирский Электротехнический институт (ныне НГТУ) по специальности «Автоматизированные системы управления» В настоящее время работает доцентом в Новосибирском государственном техническом университете, имеет более 35 лет преподавательского и научно-исследовательского стажа. В течение 2.5 лет преподавал в Kyung Hee University (Южная Корея). Член Программных комитетов и рецензент множества международных конференций, редактор журнала «Open Computer Science». Имеет около 150 публикаций. Область научных интересов: гибридные интеллектуальные системы, нейронные сети, естественный язык, умные среды, когнитивная робототехника.



Панченко Константин Евгеньевич окончил Новосибирский Государственный Технический Университет по специальности “Программное обеспечение вычислительной техники и автоматизированных систем” в 2008 году. Имеет более 8 лет опыта в сфере разработки программного обеспечения. Область научных интересов: нейроморфные технологии, нейронные сети, техническое зрение.