УДК 004.032.26

DOI: 10.15827/0236-235X.127.419-432

Дата подачи статьи: 14.01.19

2019. Т. 32. № 3. С. 419–432

Разработка метода самообучения импульсной нейронной сети для защиты от DDoS-атак

Е.В. Пальчевский¹, аспирант, teelxp@inbox.ru

О.И. Христодуло¹, д.т.н., профессор, o-hristodulo@mail.ru

¹ Уфимский государственный авиационный технический университет,
г. Уфа, 450008, Россия

Статья посвящена разработке специализированного метода обучения импульсной нейронной сети, позволяющего ускорить обнаружение и ликвидацию атак внешним несанкционированным трафиком.

Рассмотрена проблема защиты доступности информации и обучения нейронных сетей, а также обоснована необходимость проведения математического анализа для создания новых способов самообучения нейронных сетей. Представлена разработанная самообучаемая импульсная нейронная сеть, необходимая для защиты от DDoS-атак. Разработан новый метод самообучения импульсной нейронной сети, в основу которого входит равномерное распределение нейронов по всем ядрам каждого процессора в кластере. Это позволяет нейронной сети обучаться в короткие сроки с нуля (530 минут), как следствие – быстро и эффективно ликвидировать DDoS-атаки.

Проведено тестирование разработанной импульсной нейронной сети в условиях двух режимов – боевого и нормального. В результате получены нагрузочные значения на физические ресурсы каждого физического сервера в кластере. Длительное тестирование импульсной нейронной сети показывает достаточно низкую нагрузку на центральный процессор, оперативную память и твердотельный накопитель при DDoS-атаках. Соответственно, оптимальная нагрузка не только повышает доступность каждого физического сервера, но и предоставляет возможность параллельного запуска ресурсоемких вычислительных процессов без какого-либо нарушения функционирования рабочей среды.

Тестирование проводилось на серверах вычислительного кластера, где импульсная нейронная сеть показала стабильную работу и эффективно защищала от DDoS-атак.

Ключевые слова: информация, передача данных, сети, DDoS, AntiDDoS, UNIX, ОС, данные, обработка данных, информационная безопасность.

В современном мире информационных технологий одним из важнейших направлений является искусственный интеллект [1–3]. Данное направление внедряют повсеместно: начиная от сельского хозяйства и заканчивая банковским сектором [4–6]. Искусственный интеллект применяется в области информационной безопасности [7–9], например, использование нейронной сети в распознавании биометрических показателей человека или в защите от DDoS-атак [10]. Если рассматривать нейронную сеть в защите от DDoS-атак, то необходимо отметить, что данное решение является универсальным: можно разработать *искусственную нейронную сеть* (ИНС) как с обучением, так и самообучаемую. Общая схема работы нейронной сети с обучением (обучение с учителем) представлена на рисунке 1.

Для выборки входной информации при обучении нейронной сети с учителем (рис. 1) необходимо подключиться к БД. После подключения информация поступает на входной слой нейронов. На следующем шаге обучения

нейронной сети с учителем выполняется анализ данных: если обнаружена ошибка в данных, то происходит ее исправление (человеком) и возврат к шагу «прием данных»; если ошибка не обнаружена, осуществляются распространение сигнала по нейронной сети и подстройка весов (коэффициентов) для выбора приоритетного расчета данных.

Основным преимуществом методов обучения нейронной сети с учителем является человеческий фактор: экспериментатор может вмешиваться в процесс обучения – изменять приоритеты синапсов и вручную выбирать данные для обучения. Таким образом, за счет человеческого фактора появляется возможность получить именно тот результат, который необходим для достижения той или иной цели. Если рассматривать основные недостатки методов обучения с учителем, то необходимо выделить неточность результатов (в виде отклонения полученных значений от ожидаемых) и долгое время обучения нейронной сети, обусловленное зачастую тем, что человек сам подбирает

критерии расчетов, в большинстве своем не самые оптимальные.

Основными преимуществами метода обучения без учителя (самообучение нейронной сети) являются быстрдействие (достаточно быстрое обучение) и автоматический подбор нейронной сетью критериев расчета, посредством чего достигается более точный результат. Схема работы данного метода самообучения нейронной сети представлена на рисунке 2.

Необходимо отметить, что общая схема обучения нейронной сети без учителя (рис. 2) схожа со схемой обучения нейронной сети с учителем (рис. 1). Отличием являются автоматическая выборка данных, их прием и распознавание, а также автоматическое исправление ошибок в полученной исходной информации. В случае метода обучения с учителем эти действия выполняет человек. Основным недостатком данного метода обучения ИНС является ресурсоемкость. Как правило, при самообучении нейронной сети задействовано гораздо больше ресурсов физического сервера, что объясняется достаточно высокой точностью выходных результатов и быстрой обучаемостью ИНС.

В настоящее время разработкой нейронных сетей и их методов обучения для различных об-

ластей человеческой деятельности занимаются многие исследователи и ученые, например, в [11] рассмотрена искусственная нейронная сеть как средство и метод статистической обработки данных, в [12] диагностированы маршруты транспортной сети передачи данных с применением нейронной сети, в [13] рассмотрена ассоциативная спайковая нейронная сеть как один из способов обработки видеоданных, в [14] описывается распознавание английского текста сверточной нейронной сетью. Прогнозированию среднесуточной температуры воздуха нейронной сетью типа многослойный персептрон посвящена работа [15]. В [16] предложена методика подготовки данных для обработки импульсными нейронными сетями, а в [17] оценена прогнозируемость системы поддержания пластового давления нейронными сетями на нефтяных месторождениях. В работе [18] описано распознавание монохромных изображений сверточными нейронными сетями, а в [19] показано преобразование сети Петри в нейронную сеть.

Необходимо отметить, что импульсная нейронная сеть для защиты от DDoS-атак более оптимизирована под высокие нагрузки и способна обрабатывать большие объемы данных.

Функциональные спецификации и технические возможности в настоящее время позво-

ляют разрабатывать и альтернативные нейронные сети с применением нечеткой логики. Это дает потенциал не только для совершенствования существующих разработок, но и для создания полностью инновационных и уникальных продуктов в области искусственного интеллекта, которые повышают производительность серверного оборудования, снижают затраты на закупку и обслуживание физических серверов, а также автоматизируют решение каких-либо задач в различных организациях.

Целью данного научного исследования является разработка импульсной нейронной сети и метода ее самообучения для анализа, обнаружения и защиты от DDoS-атак в UNIX-подобных системах. Достижение поставленной цели позволит в автоматическом режиме

- анализировать внешний сетевой трафик для обнаружения в нем каких-либо отклонений от нормы;
- повысить доступность информации каждого сервера вычислительного кластера;
- работать с большим объемом данных;
- равномерно распределять вычислительную и сетевую загруженность по ядрам кластера;
- увеличить скорость обучения импульсной нейронной сети для защиты от DDoS-атак.

Анализ существующих методов обучения нейронных сетей

Обучение нейронной сети заключается в настройке параметров ИНС посредством моделирования той среды, в которую данная сеть планируется к внедрению или уже внедрена. Как было описано ранее, существуют два метода обучения искусственной нейронной сети: с учителем и самообучение. Например, обучение с учителем включает в себя следующие основные методы изучения заданной предметной области (задачи) нейронной сетью: метод Розенблатта, метод Уидроу–Хоффа, метод обратного распространения.

Метод Розенблатта. Схема метода обучения Розенблатта [20] при программной реализации представлена на рисунке 3.

Изначально происходит считывание выборки из предоставляемых входных данных нейронной сети. Входными данными может быть, например, информация о трафике с внешнего сетевого интерфейса. Затем входные данные конвертируются в образ x^α . Соответственно, в результате формируется выходной

Рис. 3. Алгоритм обучения нейронной сети методом Розенблатта

Fig. 3. Algorithm of training a neural network using the Rosenblatt method

образ $y^\alpha \neq y^\alpha$. После формирования выходного образа происходит подстройка начальных значений (коэффициентов) каждого синапса и как следствие – нейрона: $W(t=0)$. Далее автоматически вычисляется вектор ошибки (погрешность промежуточного значения от эталонного):

$$W(t + \Delta t) = W(t) + \eta x^\alpha \cdot (\delta^\alpha)^T, \quad (1)$$

где η – коэффициент скорости обучения; $0 < \eta < 1$ – темп обучения. Финальной частью данного алгоритма является подстройка темпа (скорости) обучения под входные данные. То есть при большом объеме входных данных скорость обучения нейронной сети замедляется из-за увеличивающегося числа итераций внутри каждого нейрона, в случае малого объема входных данных скорость обучения увеличивается. Таким образом, общая формула обучения нейронной сети методом Розенблатта будет следующей:

$$W_{ij}(t+1) = W_{ij}(t) + \eta \cdot x_i \cdot d_j, \quad (2)$$

где x_i – каждый вход нейронной сети; d_j – эталонный (желаемый) выход нейронной сети; η – коэффициент скорости обучения.

Основным преимуществом данного метода обучения является малое потребление ресурсов физического сервера нейронной сетью при сравнительно большом объеме входных данных, основным недостатком – низкая скорость обучения при большом объеме входных данных.

Метод Уидроу–Хоффа. Данный метод предполагает минимизацию среднеквадратичной ошибки ИНС [21]. Для выходных образов

минимизация среднеквадратичной ошибки нейронной сети определяется по формуле

$$E = \sum_{k=1}^L E(k) = \frac{1}{2} \sum_{k=1}^L (d_1^k - y^k)^2, \quad (3)$$

где L – количество обработанных нейронной сетью выборок данных (примеров); $E(k)$ – ошибка для k -го примера; y^k – выход нейронной сети для каждого k -го примера; d^k – эталонный (желаемый) выход нейронной сети для каждого k -го примера. Необходимо отметить, что минимизация E вычисляется на основе метода градиентного спуска:

$$w_j(t+1) = w_j(t) - a \cdot \frac{dE(k)}{dw_j(t)}, \quad (4)$$

где $\frac{dE(k)}{dw_j(t)} = \frac{dE(k)}{dy^k} \cdot \frac{dy^k}{dw_j(t)} = (y^k - d^k) \cdot x_j^k$. Таким

образом, изменение весовых коэффициентов нейронной сети происходит по следующему правилу:

$$w_j(t+1) = w_j(t) - a \cdot (y^k - d^k) \cdot x_j^k. \quad (5)$$

На рисунке 4 представлен алгоритм обучения с учителем Уидроу–Хоффа (дельта-правило) при программной реализации.

Основными преимуществами обучения нейронной сети методом Уидроу–Хоффа являются возможность обработки больших объемов данных за одну итерацию и высокая точность получаемых результатов на выходе.

Если рассматривать недостатки данного метода обучения нейронной сети, то в качестве основного необходимо выделить проблему вы-

бора значения шага обучения: если шаг слишком мал, обучение нейронной сети занимает долгое время; если шаг обучения большой, процесс обучения может оказаться расходящимся, как следствие – поставленная задача может быть не решена.

Метод обратного распространения ошибки. Нейронную сеть, имеющую два или более слоев, невозможно обучить методами Розенблатта и Уидроу–Хоффа. Причиной возникновения данной проблемы являются скрытые слои в ИНС. Таким образом, в многослойных нейронных сетях известен лишь выход последнего слоя нейронов [22]. Решением данной проблемы как раз и является метод обратного распространения.

Основная идея обучения нейронной сети методом обратного распространения ошибки состоит в распространении сигналов ошибки от выходов ИНС к ее входам. Изначально ставится задача минимизации функции ошибки:

$$E = \frac{1}{2} \sum_{j=1}^p (y_j - d_j)^2, \quad (6)$$

где y_j – получаемое значение j -го выхода ИНС; d_j – эталонное значение j -го выхода нейронной сети. Соответственно, минимизация E определяется методом градиентного спуска. Изначально происходит автоматическая подстройка весовых коэффициентов синапсов:

$$\Delta w_{ij} = -h \cdot \frac{dE}{dw_{ij}}, \quad (7)$$

где w_{ij} – вес синапсов; $-h$ ($0 < h < 1$) – скорость обучения нейронной сети, выражаемая коэффициентом. Далее необходимо раскрыть (7) для более точного обучения нейронной сети.

Таким образом,

$$\frac{dE}{dw_{ij}} = \frac{dE}{dy_j} \cdot \frac{dy_j}{ds_j} \cdot \frac{ds_j}{dw_{ij}}, \quad (8)$$

где s_j – сумма входных сигналов каждого нейрона ИНС; y_j – выход j -го нейрона.

Соответственно,

$$\frac{dE}{dy_j} = \sum_k \frac{dE}{dy_k} \cdot \frac{dy_k}{ds_k} \cdot \frac{ds_k}{dy_j} = \sum_k \frac{dE}{dy_k} \cdot \frac{dy_k}{ds_k} \cdot w_{jk}^{[n+1]}, \quad (9)$$

где k – количество нейронов в слое $n + 1$.

Также для раскрытия (7) вводим новую переменную:

$$\delta_j^{[n+1]} = \frac{dE}{dy_j} \cdot \frac{dy_j}{ds_j}. \quad (10)$$

После ввода переменной получаем рекурсивные формулы соответственно для выходного и входного слоев ИНС:

$$\delta_i^{[n]} = \frac{dy_i}{ds_i} \cdot (y_i^{[n]} - d_i), \quad (11)$$

$$\delta_j^{[n]} = \frac{dy_j}{ds_j} \cdot \sum_k \delta_k^{[n+1]} \cdot w_{jk}^{[n+1]}. \quad (12)$$

Таким образом, раскрытая формула (7) автоматической подстройки весовых коэффициентов синапсов будет следующей:

$$\Delta w_{ij} = -h \cdot \delta_j^{[n]} \cdot y_i^{[n-1]}. \quad (13)$$

Схема данного алгоритма обучения представлена на рисунке 5. Изначально ИНС считывает входную информацию из БД. Далее для формирования сигналов в промежуточных слоях многослойной нейронной сети в случайном порядке распределяются коэффициенты (веса) синапсов. Следующим шагом является минимизация (вычисление) ошибки E (6): если полученное значение отклоняется от нормы, то происходит возврат к выборке данных (как правило, разработчик или тестировщик исправляет входную информацию). В случае приемлемого отклонения полученного значения от эталонного происходит проверка заданных ранее критериев итераций в нейронах. Заданная выборка прогоняется по нейронной сети. После первого прогона выборки по ИНС автома-

тически подстраиваются коэффициенты весов синапсов. Далее происходит повторный прогон данных по нейронной сети для получения более точного результата.

Основные преимущества обучения нейронной сети методом обратного распространения ошибки:

- автоматический контроль усиления сигналов: для слабых сигналов (когда выходное значение близко к нулю) синапсы имеют больший коэффициент веса; таким образом, чем больше вес синапса, тем сильнее сигнал;
- автоматическая подстройка весов синапсов для повышения точности результатов.

Основные недостатки данного метода обучения нейронной сети:

- размер шага коррекции весов синапсов; в случае автоматической подстройки весов синапсов их значения могут стать как слишком большими, так и минимальными, а это замедляет обучение нейронной сети;
- большое потребление ресурсов физического сервера; данный недостаток вытекает из предыдущего: чем дольше нейронная сеть обучается, тем больше серверных ресурсов она потребляет.

Если рассматривать самообучение (без учителя), то необходимо выделить один из основных способов обучения нейронной сети – метод Хебба.

Метод Хебба. Данный метод самообучения нейронной сети основывается на подстройке весов синапсов [23]. Существуют две разновидности метода обучения Хебба: сигнальный и дифференциальный. Сигнальный метод обучения Хебба основан на усилении связей между возбужденными нейронами. Таким образом, веса синапсов будут распределяться следующим образом:

$$w_{ij}(t+1) = w_{ij}(t) + a \cdot y_i^{[n-1]} \cdot y_j^{[n]}, \quad (14)$$

где a – скорость обучения (коэффициент); $y_i^{[n-1]}$ – значение на выходе у каждого i -го нейрона слоя $n-1$; $w_{ij}(t+1)$ и $w_{ij}(t)$ – весовые коэффициенты синапса, который соединяет нейроны на итерациях t и $t+1$; $y_j^{[n]}$ – выходное значение каждого j -го нейрона слоя n .

Алгоритм обучения нейронной сети сигнальным методом Хебба идентичен тому, что показан на рисунке 2. Единственным отличием является автоматическая подстройка весов синапсов сразу же после приема входных данных. Это позволяет ИНС быстрее обучаться и минимизировать ошибки при обработке информации.

Если рассматривать дифференциальный метод обучения Хебба, то необходимо отметить, что он основан (как и сигнальный) на обучении синапсов, соединяющих те нейроны ИНС, входы которых наиболее успешно динамично изменились в сторону увеличения. Таким образом, формула обучения будет следующей:

$$w_{ij}(t+1) = w_{ij}(t) + a \cdot ((y_i^{[n-1]}(t) - y_i^{[n-1]}(t-1)) \times (y_j^{[n]}(t) - y_j^{[n]}(t-1))), \quad (15)$$

где $y_j^{[n]}(t)$ и $y_j^{[n]}(t-1)$, а также $y_i^{[n-1]}(t)$ и $y_i^{[n-1]}(t-1)$ – выходные значения каждого i -го и j -го нейронов слоев n и $n-1$ на итерациях t и $t-1$.

Также необходимо рассмотреть алгоритм дифференциального метода обучения Хебба, состоящий из следующих этапов:

- 1) считывание входных данных (выборки);
- 2) присвоение всем весовым коэффициентам синапсов значений малой величины;
- 3) формирование входного образа;
- 4) распространение сигналов возбуждения по всем слоям нейронной сети – для каждого нейрона рассчитывается сумма его входов;
- 5) изменение весовых коэффициентов на основании полученных выходных значений;
- 6) полное повторение всех этапов (начиная со 2-го и заканчивая 5-м) до тех пор, пока выходное значение нейронной сети не стабилизируется с заданной точностью;
- 7) нейронная сеть обучена.

Основными преимуществами данных методов обучения Хебба являются универсальность, самостоятельность и точность. Универсальность сигнального и дифференциального методов обучения Хебба заключается в возможности их применения в различных видах ИНС, например, когнитрон или неокогнитрон. Под самостоятельностью подразумевается самообучение: нейронная сеть обучается без участия человека, а под точностью – самообучение нейронной сети до тех пор, пока не будет достигнуто необходимое значение.

В качестве основных недостатков данных методов обучения необходимо выделить ресурсоемкость и скорость. Во время обучения данными ме-

тодами нейронная сеть потребляет достаточно большое количество ресурсов физического сервера из-за точности обучения. Вторым основным недостатком также следует из точности обучения: чем точнее должно быть значение на выходе, тем дольше нейронная сеть будет обучаться.

Приведенные аналогичные методы обучения нейронных сетей используются достаточно широко по следующим причинам: простота в реализации и необходимый базовый функционал (при программной реализации ИНС) для решения простых задач. Существенным минусом всех вышеприведенных решений является невозможность их использования для решения каких-либо сложных и емких задач, например, обучение нейронной сети для обработки и хранения больших данных в облачном хранилище или отражение емких массивных DDoS-атак. Таким образом, разработка новых методов обучения нейронной сети является важной и актуальной задачей.

Разработка и реализация самообучаемой импульсной нейронной сети для защиты от DDoS-атак

Необходимо отметить, что импульсная (спайковая) нейронная сеть представляет собой ИНС, в которой нейроны обмениваются короткими импульсами одинаковой амплитуды. Общая структура спайковой нейронной сети представлена на рисунке 6. Схема разработанной спайковой нейронной сети для защиты от DDoS-атак показана на рисунке 7.

Изначально считываются данные с внешнего сетевого интерфейса (рис. 7). Далее в автоматическом режиме осуществляется кодирование информации в сигналы по методу синхронного кодирования (рис. 8). После кодирования информации происходит генерация импульсов нейронами. Далее импульсы с одинаковой амплитудой и длительностью передаются на входной слой и происходит обработка импульсов (обучение ИНС) внутри промежуточных слоев с последующим кодированием в получаемое значение на выходе.

Как правило, при синхронном кодировании в спайковых нейронных сетях эталоном считается одновременная генерация спайков нейронами, но допускается некоторая рассинхронизация в рамках временного окна (рис. 8).

Основные преимущества разработанной спайковой нейронной сети. Необходимо отметить, что зачастую импульсные нейронные сети обучаются без учителя. Разработанный способ обучения импульсной ИНС также относится к этому методу. Алгоритм обучения разработанной импульсной нейронной сети показан на рисунке 9.

Первый этап. Изначально нейронная сеть считывает данные с внешнего сетевого интерфейса. Под данными подразумеваются скорость внешнего сетевого

интерфейса (Гбит/с), количество входящих сетевых пакетов, IP-адреса, текущие правила стандартных средств защиты доступности данных. Например, IPTables в UNIX-подобных системах. Далее происходит преобразование информации в сигналы методом синхронного кодирования:

$$S_q = k \cdot (V_q - V_b) \cdot (V_q - V_t) - U_q + I_{syn}, \quad (16)$$

где g – количество входящих сетевых пакетов; k – коэффициент преобразования; S_q – количество импульсов на каждый слой нейронной сети; V_q – скорость сетевого интерфейса; V_b – количество IP-адресов, с которых поступает сетевой трафик; V_t – время сетевой активности (сколько времени каждый физический сервер кластера находится во внешней глобальной сети); I_{syn} – синаптический входной ток. Соответственно, если $(V_q + g) \geq (V_q + V_t)$, то

$$\begin{cases} V_q = k, \\ U_q = dV_t. \end{cases} \quad (17)$$

Второй этап. Далее импульсы передаются на входной слой, где происходит их считывание для дальнейшего обучения:

$$D(s) = \sum_{ij}^n \sum_{ij}^m S_q \cdot (t - t_j^i), \quad (18)$$

где n – количество входных каналов каждого i -го нейрона j -го (в данном случае – входного) слоя; m – количество выходов каждого нейрона i -го нейрона j -го слоя; t – общая входная последовательность импульсов; t_j^i – входная последовательность импульсов на каждый i -й нейрон j -го слоя.

Далее происходит подсчет импульсов на входном слое ИНС:

$$F_{ij}(t) = \sum_{D(s)} \frac{1}{\lambda \cdot D(s)} + \sum_{D(s)} \int \frac{dh}{du}, \quad (19)$$

где $F_{ij}(t)$ – количество импульсов, образованных на входном слое спайковой нейронной сети; $D(s)$ – количество считанных импульсов; λ – плотность каждого импульса; h – общее время генерации дополнительных импульсов на входном слое ИНС.

После подсчета импульсов необходимо определить количество промежуточных слоев в нейронной сети:

$$T_{ij} = F_{ij}(t) \cdot \frac{1}{2} C \cdot \frac{1}{2} R, \quad (20)$$

где T_{ij} – количество промежуточных слоев в нейронной сети; C – загруженность процессоров кластера; R – загруженность оперативной памяти кластера. Необходимо отметить, что, чем больше свободных ресурсов на кластере, тем больше нейронная сеть генерирует промежуточных слоев. Соответственно, это увеличит точность выходных данных. Также в каждом промежуточном слое создается определенное количество нейронов (зависит от количества созданных промежуточных слоев):

$$NE_{ij} = F_{ij}(t) \cdot T_{ij}, \quad (21)$$

где NE – количество i -х нейронов в каждом j -м (промежуточном) слое.

Третий этап. Рассчитываем количество необходимых ресурсов кластера для полноценной работы нейронной сети. Изначально необходимо определить среднее количество входящего сетевого трафика за один календарный месяц:

$$Z = \frac{1}{12} \cdot y \cdot \sum_{v_q}^g \sum_{v_b}^{V_i} ((x_i \cdot x_j) + (V_q - 1)), \quad (22)$$

где Z – среднее количество входящего сетевого трафика за один календарный месяц; y – прошедшие 12 календарных месяцев; x_i – количество легитимного трафика за последний месяц; x_j – количество вредоносного трафика за последний месяц.

Также необходимо узнать теоретическую сетевую нагрузку на ресурсы кластера во время атак внешним несанкционированным трафиком:

$$\frac{dV_q}{dV_t} = -\sum_{Z=1} (V_h - Z) \cdot \frac{dC}{dR} = -\sum_{Z=1} (V_h - Z) \cdot C'(Z). \quad (23)$$

Введем обозначение:

$$\Delta_j^z = (V_h - Z) \cdot C'(Z) = (V_h - Z) \cdot C'(Z). \quad (24)$$

Соответственно, выражение (23) можно преобразовать следующим образом:

$$\frac{dV_q}{dV_t} = -\sum_{Z=1} \Delta_j^z C'(Z). \quad (25)$$

Так как $C'(Z) = 1$, выражение (23) примет следующий вид:

$$\frac{dV_q}{dV_t} = (V_h - Z). \quad (26)$$

Таким образом, количество необходимых ресурсов кластера для полноценной работы (защиты от DDoS-атак) импульсной нейронной сети рассчитывается по формуле

$$K_r = \frac{1}{2} \cdot ((T_{ij} + F_{ij}(t)) \cdot (V_h - Z)), \quad (27)$$

где K_r – необходимые ресурсы кластера. Выражается в виде процентной загруженности (не больше 50 % или 1/2 от свободных мощностей кластера) центральных процессоров и оперативной памяти.

Следующим шагом обучения импульсной нейронной сети является равномерное распределение всех слоев и нейронов ИНС по процессорным ядрам кластера:

$$R_m = \sum_c \frac{(F_s \cdot C_1 \cdot C \cdot M_g)}{\frac{1}{2} \cdot M_g}, \quad (28)$$

где R_m – коэффициент равномерного распределения (на его основе и происходит перенаправление нейронов по слоям и слоев по ядрам каждой ЭВМ кластера); F_s – количество физических серверов в кластере; C_1 – количество процессоров в кластере; M_g – количество ядер в кластере.

Далее происходит подготовка итераций внутри нейронов ИНС. Соответственно, проверяются значения каждого импульса. Каждый импульс должен быть равен состоянию «1». Необходимо отметить, что только при таком

состоянии происходит возбуждение нейрона для обработки данных и передачи сигналов. В случае, если импульс $<$ или $>$ 1, происходит генерация новых импульсов до тех пор, пока каждый спайк не будет равен 1.

Следующий подэтап обучения нейронной сети – первичный и повторный прогоны импульсов в промежуточных слоях. Это необходимо для проверки корректности сконвертированных данных методом синхронного кодирования.

Четвертый этап. На данном этапе происходит подача импульсов на выходной слой:

$$D_1(s) = \sum_{ij}^n \sum_{ij}^m S_{iq} \cdot (t_1 + t_{1j}^i), \quad (29)$$

где $D_1(s)$ – скорость подачи импульсов на выходной слой; S_{iq} – количество импульсов на выходе каждого нейрона; t_1 – общая выходная последовательность импульсов; t_{1j}^i – выходная последовательность импульсов каждого i -го нейрона j -го слоя.

Пятый этап. Необходимо преобразовать импульсы в информацию для получения читабельного выходного значения:

$$S_q = k \cdot S_{iq}. \quad (30)$$

Таким образом, реализованный метод обучения импульсной нейронной сети позволяет ускорить процесс выявления DDoS-атак и в кратчайшие сроки ликвидировать данные атаки, тем самым снижая риск простоя серверного оборудования.

Апробация самообучаемой импульсной нейронной сети для защиты от DDoS-атак

После разработки импульсной нейронной сети для защиты от DDoS-атак проводится тестирование, необходимое для выявления эффективности предлагаемого решения. С целью более точного определения значений средней нагрузки в календарный день данные вынесены в отдельные таблицы.

Потребление вычислительных ресурсов без каких-либо атак сетевым трафиком и при активированном состоянии импульсной нейронной сети (в течение 10 календарных дней) представлено в таблице 1 (тестирование в обычном режиме).

Исходя из данных таблицы 1, средняя нагрузка на физические ресурсы кластера импульсной нейронной сетью следующая:

- центральный процессор (CPU): 0,39 %;
- оперативная память (ОЗУ): 1,50 %;
- твердотельный накопитель (SSD): 5,53 %.

Таким образом, загруженность ресурсов ЭВМ достаточно низкая и не вызывает каких-либо перегрузок всего кластера. Также необходимо отметить, что во время тестирования в обычном режиме все серверы доступны по внешней глобальной сети в связи с отсутствием DDoS-атак.

Далее необходимо провести тестирование в «боевом» режиме. Потребление вычислительных ресурсов при массивных DDoS-атаках и при активированном состоянии импульсной нейронной сети (в течение 10 календарных дней) представлено в таблице 2.

Исходя из данных таблицы 2, средняя нагрузка на ресурсы ЭВМ будет следующей:

- центральный процессор (CPU): 12,92 %;
- оперативная память (ОЗУ): 10,62 %;
- твердотельный накопитель (SSD):

14,50 %.

Таким образом, нагрузка на ресурсы всех физических серверов кластера достаточно низкая, несмотря на массивные DDoS-атаки. Также необходимо отметить, что при DDoS-атаках повышалось количество итераций и нейронов ИНС. Это связано с тем, что при по-

ступлении большого количества внешнего сетевого трафика генерируются дополнительные нейроны и, как следствие, увеличивается общее количество итераций в импульсной нейронной сети. Все физические серверы во время DDoS-атак были доступны по внешней глобальной сети и работали без перебоев за счет быстрого реагирования, обучения и ликвидации атак типа DDoS импульсной нейронной сетью.

Сравнение методов обучения импульсных нейронных сетей

В рамках апробации произведено сравнение (табл. 3) методов обучения ИНС (разработанный метод обучения импульсной нейронной сети и аналоги). Данный анализ необходим для выявления эффективности разработанного аппаратно-программного модуля.

Предложенный метод обучения показал следующее:

- достаточно высокую скорость (600 минут на полное обучение) при самообучении для защиты от DDoS-атак;

Таблица 1

Загруженность ресурсов всех ЭВМ кластера в обычном режиме

Table 1

Resource load on all cluster computers in normal mode

Характеристика	День									
	1	2	3	4	5	6	7	8	9	10
Количество итераций ИНС, тыс. шт./с	0,80	2,14	3,67	5,18	6,23	7,29	8,65	9,00	10,12	11,00
Скорость внешнего сетевого интерфейса, Гбит/с	0,30	0,40	0,50	0,60	0,70	0,80	0,90	1,00	1,10	1,20
Количество нейронов ИНС, тыс. шт.	0,70	0,80	0,90	1,00	1,10	1,20	1,30	1,40	1,50	1,60
Нагрузка на CPU, %	0,14	0,19	0,23	0,27	0,34	0,37	0,45	0,50	0,68	0,79
Потребление ОЗУ, %	0,32	0,58	0,90	1,12	1,32	1,59	1,78	2,00	2,54	2,85
Нагрузка на SSD, %	1,00	2,08	3,03	4,09	5,10	6,00	7,00	8,01	9,03	10,00
Доступность серверов	Да									

Таблица 2

Загруженность ресурсов всех ЭВМ кластера в «боевом» режиме

Table 2

The workload of all cluster computer resources in the combat mode

Характеристика	День									
	1	2	3	4	5	6	7	8	9	10
Количество итераций ИНС, тыс. шт./с	36,80	38,14	40,80	42,96	44,80	46,49	48,89	50,90	52,80	54,99
Скорость DDoS-атаки, Гбит/с	8,00	9,00	10,00	11,00	12,00	13,00	14,00	15,00	16,00	17,00
Количество нейронов ИНС, тыс. шт.	18,00	19,00	20,00	21,00	22,00	23,00	24,00	25,00	26,00	27,00
Нагрузка на CPU, %	8,00	9,12	10,17	11,18	12,15	13,19	14,80	15,76	16,90	17,98
Потребление ОЗУ, %	6,00	7,00	8,01	9,07	10,00	11,00	12,00	13,13	14,26	15,76
Нагрузка на SSD, %	10,00	11,00	12,00	13,00	14,00	15,00	16,00	17,00	18,00	19,00
Доступность серверов	Да									

Сравнение методов обучения импульсной нейронной сети
Comparison of training methods of a spiking neural network

Таблица 3
Table 3

Параметр	Общее обучение без учителя	Глубокое обучение	Метод ReSuMe	Обучение обратным распространением ошибки	Метод обучения авторов статьи
Скорость обучения с нуля при DDoS-атаках, мин.	600,00	800,00	750,00	670,00	530,00
Максимально возможное количество обрабатываемых данных перед кодированием в импульсы (зависит от алгоритма обучения и оборудования: чем оно мощнее, тем больше данных можно обработать; в рассматриваемом случае сравнение происходит на оборудовании, характеристика которого также представлена в данном разделе статьи), Гбит/с	10,00	12,00	11,65	10,54	13,19
Возможность обучения нейронной сети с расчетом текущей занятости вычислительных ресурсов	Нет				Да
Возможность равномерного распределения слоев и нейронов по всем ядрам сервера/кластера	Нет				Да

– возможность обработки большего объема данных (13.19 Гбит/с) по сравнению с остальными рассматриваемыми методами обучения;

– возможность расчета вычислительных ресурсов кластера для самообучения импульсной нейронной сети, что позволяет не перегружать ресурсы кластера и параллельно запускать другие ресурсоемкие процессы, необходимые для решения тех или иных задач;

– возможность равномерного распределения слоев и нейронов по всем ядрам сервера/кластера; данный параметр также позволяет снизить загруженность ресурсов кластера, что повышает его производительность и предоставляет возможность запуска других ресурсоемких систем для решения каких-либо задач.

Также необходимо отметить, что разработанная нейронная сеть была распараллелена по ядрам процессоров каждого физического сервера в кластере. Это позволяло не только дополнительно снизить общую загруженность кластера, но и повысить производительность всей импульсной нейронной сети.

Тестирование разработанной импульсной нейронной сети (при DDoS-атаках и большом количестве обрабатываемой информации) проводилось на следующем оборудовании:

– количество физических серверов – 30;

– процессор Intel Xeon 5690 (CPU – 60, физических ядер – 360, количество потоков – 720);

– оперативная память – 960 ГБ;

– SSD – RAID 10 (Samsung 850 pro 1 ТБ каждый);

– внешний сетевой канал – 20 Гбит/с;

– внутренний сетевой канал (локальная сеть) – 100 Гбит/с.

Данное оборудование соответствует современным требованиям и позволяет обрабатывать большие объемы данных.

Заключение

В ходе проведенных исследований были получены следующие результаты.

Предложен метод обучения импульсной нейронной сети, позволяющий реализовать структуру самообучения (без учителя) ИНС на программном уровне. Для обучения информация преобразуется методом синхронного кодирования в импульсы (спайки). На выход подаются также преобразованные импульсы, но уже в читабельном виде (получаем готовые правила фильтрации DDoS-атаки).

Разработана импульсная нейронная сеть, предназначенная для отражения DDoS-атак в кратчайшие сроки. Самообучение происходит с помощью предложенного метода обучения без учителя. На вход подаются следующие данные: количество входящих сетевых пакетов, IP-адреса, текущие правила стандартных средств защиты доступности данных.

Проведено множество тестирований, доказывающих целесообразность использования разработанных импульсной нейронной сети и метода самообучения ИНС. Средняя нагрузка на центральный процессор варьируется от 0,39 до 12,92 %. Достаточно низкая нагрузка на CPU позволяет запускать многочисленные ресурсоемкие процессы как в «боевом» (во время DDoS-атаки), так и в обычном режимах без ка-

ких-либо потерь производительности. Средняя загруженность SSD-накопителей колеблется от 5,53 до 14,50 %. Столь небольшая нагрузка на твердотельный накопитель предоставляет возможность использования ресурсов SSD другими ресурсоемкими процессами. Среднее потребление ресурсов оперативной памяти варьируется в диапазоне 1,50–10,62 %. Небольшое потребление ресурсов ОЗУ также говорит о целесообразности использования разработанных импульсной нейронной сети и метода самообучения ИНС.

Таким образом, низкая нагрузка на вычислительные ресурсы позволяет не только в кратчайшие сроки защитить каждую ЭВМ кластера от DDoS-атак, но и повысить их доступность наряду с производительностью.

Литература

1. Шумихин А.Г., Бояршинова А.С. Параметрическая идентификация управляемого объекта в режиме его эксплуатации с применением технологии нейронных сетей // Вестн. ПНИПУ Электротехника, информационные технологии, системы управления. Пермь: Изд-во ПНИПУ, 2016. № 19. С. 102–110.
2. Мелихова О.А., Вепринцева О.А., Чумичев В.С., Джамбинов С.В., Гайдуков А.Б. Режимы обучения в искусственных нейронных сетях // Инновации в науке. 2016. № 53. С. 16–23.
3. Королев Д.С., Токан В.В. Определение взрывопожароопасных зон, основываясь на молекулярных дескрипторах и искусственных нейронных сетях // Пожарная безопасность: проблемы и перспективы. 2016. Т. 2. № 1. С. 279–281.
4. Кольчери́н Д.В., Печеркин С.А. Применение мультиагентных систем, основанных на нейронных сетях // Проблемы современной науки и образования. 2017. № 23. С. 6–10.
5. Воевода А.А., Романников Д.О. Пример реализации выбора минимального числа в бинарном виде на нейронных сетях // Сб. науч. тр. НГТУ. 2017. № 1. С. 64–71.
6. Игнатенков А.В., Ольшанский А.М. Управление величиной ошибки в нейронных сетях // Изв. СамНЦ РАН. 2016. Т. 18. № 4–4. С. 733–738.
7. Агян К.Ю., Ханжин В.Г. Нейронная сеть с архитектурой Кохонена для распознавания изображений // Прочность неоднородных структур ПРОСТ: сб. тр. IX Евраз. науч.-практич. конф. М.: Студио-Принт, 2018. С. 153.
8. Убский Д.С. Сверточная нейронная сеть для распознавания речи // Альманах науч. работ молод. ученых ун-та ИТМО: матер. XLV учеб.-метод. конф. ИТМО. 2016. С. 127–128.
9. Пальчевский Е.В., Халиков А.Р. Нейронная сеть как самообучаемый фильтр от атак типа «DoS» и «DDoS» // Инновации в науке и практике: сб. матер. IV Междунар. науч.-практич. конф. Барнаул, 2017. С. 69–75.
10. Пальчевский Е.В., Халиков А.Р. Нейронная сеть – как элемент автоматизированной защиты от DDoS-атак // Актуальные вопросы современной науки: сб. матер. VII Междунар. науч.-практич. конф. Минск, 2017. С. 83–88.
11. Бондарев В.Ю., Сорокин А.С., Кротова Е.Л. Искусственная нейронная сеть как средство и метод статистической обработки данных // Вестн. УрФО. Безопасность в информационной сфере. 2016. № 2. С. 19–22.
12. Ануфренко А.В., Канаев А.К., Сахарова М.А. Диагностирование маршрутов транспортной сети передачи данных с применением нейронных сетей // Междунар. конф. по мягким вычислениям и измерениям: сб. матер. СПб, 2017. С. 231–234.
13. Борисов В.В., Чандер А.П. Ассоциативная спайковая нейронная сеть для обработки видеоданных // Нейрокомпьютеры и их применение: тез. докл. конф. М., 2016. С. 54-а.
14. Федорова А.А. Распознавание английского текста сверточной нейронной сетью // Молодой ученый. 2016. № 14. С. 97–102.
15. Худобердина Е.С., Лезин И.А. Прогнозирование среднесуточной температуры воздуха нейронной сетью типа многослойный персептрон // XIV Королёвские чтения: тез. Междунар. молодеж. науч. конф. Самара, 2017. С. 110.

16. Терехов В.И., Жуков Р.В. Методика подготовки данных для обработки импульсными нейронными сетями // Нейрокомпьютеры: разработка, применение. 2017. № 2. С. 31–36.
17. Хусаинов А.Т. Оценка прогнозируемой системы поддержания пластового давления нейронными сетями на нефтяных месторождениях // Академический журнал Западной Сибири. 2016. Т. 12. № 3. С. 48.
18. Хамидуллин А.Л., Новикова С.В. Распознавание монохромных изображений сверточными нейронными сетями // Молодежный научный форум: технические и математические науки. 2017. № 6. С. 176–182.
19. Романников Д.О. О преобразовании сети Петри в нейронную сеть: сб. науч. тр. НГТУ. 2016. № 4. С. 98–103.
20. Михов Е.О., Михова Е.Д., Непомнящий О.В. Сравнительный анализ непараметрических алгоритмов на примере моделирования стохастических процессов // Вычислительные технологии. 2017. Т. 22. № 1. С. 37–47.
21. Маркин Е.И., Подопригора И.А., Зоткина А.А., Бершадская Е.Г. Методы разработки искусственных нейронных сетей // Вестн. современных исследований. 2018. № 3. С. 49–52.
22. Смирнов Е.М., Фомин В.В. Многопоточное обучение нейронных сетей методом обратного распространения ошибки // Инновации на основе информационных и коммуникационных технологий. 2014. № 1. С. 244–246.
23. Тимофеева О.П., Палицына С.С. Обучение нейронной сети интеллектуальной системы управления городскими светофорами // Тр. НГТУ. 2016. № 1. С. 25–31.

Software & Systems
DOI: 10.15827/0236-235X.127.419-432

Received 14.01.19
2019, vol. 32, no. 3, pp. 419–432

Developing a self-learning method for a spiking neural network to protect against DDoS attacks

*E.V. Palchevsky*¹, Postgraduate Student, *teelp@inbox.ru*

*O.I. Khristodulo*¹, Dr.Sc. (Engineering), Professor, *o-hristodulo@mail.ru*

¹ Ufa State Aviation Technical University, Ufa, 450008, Russian Federation

Abstract. The paper is devoted to the development of a specialized training method for a spiking neural network, which allows speeding up the detection and elimination of attacks by external unauthorized traffic.

The paper considers the problem of protecting information availability and teaching neural networks. It also justifies the need for mathematical analysis to develop new methods of self-learning of neural networks. The paper introduces the developed self-learning spiking neural network that is necessary to protect against DDoS attacks. A new self-learning method for a spiking neural network is based on the uniform distribution of neurons across all cores of each processor in a cluster. This allows a neural network to learn from scratch in a short time (530 minutes). As a result, it quickly and effectively eliminates DDoS attacks.

The authors tested the developed spiking neural network in two modes: combat and normal. The tests gave load values for physical resources of each physical server in a cluster. Long-term testing of a spiking neural network shows fairly low load on a central processor, RAM and solid-state drive during DDoS attacks. Naturally, optimal load increases the availability of each physical server, and makes it possible to simultaneously run resource-intensive computational processes without any disruption of the working environment.

Testing was conducted on computing cluster servers in one of Moscow data centers. The spiking neural network has shown stable operation and effective protection against DDoS attacks.

Keywords: information, data transfer, networks, DDoS, AntiDDoS, UNIX, OS, data, data processing, information security.

References

1. Shumikhin A.G., Boyarshinova A.S. Parametrical identification of a managed object in the operation mode using a neural network technology. *PNRPU Bulletin. Electrotechnics, Informational Technologies, Control Systems*. Publ. "PNIPU", Perm, 2016, no. 19, pp. 102–110 (in Russ.).
2. Melikhova O.A., Veprintseva O.A., Chumichev V.S., Dzhambinov S.V., Gaidukov A.B. Training modes in artificial neural networks. *Innovations in Science*. Novosibirsk, 2016, no. 53, pp. 16–23 (in Russ.).
3. Korolev D.S., Tokan V.V. Determination of explosion and fire hazard zones based on molecular descriptors and artificial neural networks. *Fire Safety: Problems and Prospects*. Voronezh, 2016, vol. 2, no. 1, pp. 279–281 (in Russ.).

4. Kolcherin D.V., Pecherkin S.A. The use of multi-agent systems based on neural networks. *Problems of Modern Science and Education*. Ivanovo, 2017, no. 23, pp. 6–10 (in Russ.).
5. Voevoda A.A., Romannikov D.O. An example of choosing a minimum number in a binary form in neural networks. *Proc. of the NSTU*. 2017, no. 1, pp. 64–71 (in Russ.).
6. Ignatenkov A.V., Olshansky A.M. Error control in neural networks. *Proc. of the Samara Sci. Center of RAS*. 2016, vol. 18, no. 4–4, pp. 733–738 (in Russ.).
7. Agyan K.Yu., Khanzhin V.G. Kohonen's neural network for image recognition. *Proc. 9th Eurasian Sci. and Pract. Conf. "Durability of Non-Uniform Structures" (PROST 2018)*. Moscow, Studio-Print Publ., 2018, p. 153.
8. Ubsky D.S. Convolutional neural network for speech recognition. *Almanac of Scientific Works of Young Scientists of ITMO Univ. Proc. 15th Academic Conf. of the ITMO Univ*. 2016, pp. 127–128 (in Russ.).
9. Palchevsky E.V., Khalikov A.R. A neural network as a self-learning filter against DoS and DDoS attacks. *Innovations in Science and Practice: Proc. 4th Intern. Sci. and Pract. Conf.* 2017, pp. 69–75 (in Russ.).
10. Palchevsky E.V., Khalikov A.R. A neural network as an element of automated protection against DDoS attacks. *Proc. 7th Intern. Sci. and Pract. Conf. "Relevant Issues of Modern Science"*. Minsk, 2017, pp. 83–88 (in Russ.).
11. Bondarev V.Yu., Sorokin A.S., Krotova E.L. An artificial neural network as a tool and a method for statistical data processing. *UrFR Newsletter. Information Security*. 2016, no. 2, pp. 19–22 (in Russ.).
12. Anufrenko A.V., Kanaev A.K., Sakharova M.A. Diagnostics of data transport network routes using neural networks. *Proc. Intern. Conf. on Soft Computing and Measurements*. St. Petersburg, 2017, pp. 231–234 (in Russ.).
13. Borisov V.V., Chander A.P. An associative spike neural network for video data processing. *Neurocomputers and Their Application: Abstracts*. A.I. Galushkina, A.V. Chechkina, L.S. Kuravsky, S.L. Artemenkova, G.A. Yuriev, P.A. Marmalyuk, A.V. Gorbatova, S.D. Kulik (Eds.). 2016, pp. 54 (in Russ.).
14. Fedorova A.A. Recognition of the English text by a convolutional neural network. *Young Scientist*. 2016, no. 14, pp. 97–102 (in Russ.).
15. Khudoberdina E.S., Lezin I.A. Forecasting the average daily air temperature by a multilayer perceptron neural network. *Proc. Intern. Youth Sci. Conf. 14th Korolev Readings*. 2017, p. 110 (in Russ.).
16. Terekhov V.I., Zhukov R.V. Methods of preparing data for processing by spiking neural networks. *Neurocomputers: Development, Application*. 2017, no. 2, pp. 31–36 (in Russ.).
17. Khusainov A.T. Evaluation of a predicted system for maintaining reservoir pressure by neural networks in oil fields. *Academic J. of Western Siberia*. 2016, vol. 12, no. 3, p. 48 (in Russ.).
18. Khamidullin A.L., Novikova S.V. Recognition of monochrome images by convolutional neural networks. *Youth Scientific Forum: Tech. and Math. Sciences*. 2017, no. 6, pp. 176–182 (in Russ.).
19. Romannikov D.O. On the transformation of Petri nets into a neural network. *Trans. of Sci. Papers of the Novosibirsk State Tech. Univ*. 2016, no. 4, pp. 98–103 (in Russ.).
20. Mikhov E.O., Mikhova E.D., Nepomnyashchy O.V. Comparative analysis of non-parametric algorithms on the example of modeling stochastic processes. *Computing Technologies*. 2017, vol. 22, no. 1, pp. 37–47 (in Russ.).
21. Markin E.I., Podoprighora I.A., Zotkina A.A., Bershadskaya E.G. Development methods of artificial neural networks. *The Bulletin of Modern Research*. 2018, no. 3, pp. 49–52 (in Russ.).
22. Smirnov E.M., Fomin V.V. Multi-threaded learning of neural networks using the method of back-propagation of errors. *Innovations Based on Information and Communication Technologies*. 2014, no. 1, pp. 244–246 (in Russ.).
23. Timofeeva O.P., Palitsyna S.S. Training a neural network of an intelligent city traffic lights control system. *Proc. of the NSTU*. 2016, no. 1, pp. 25–31 (in Russ.).

Для цитирования

Пальчевский Е.В., Христодуло О.И. Разработка метода самообучения импульсной нейронной сети для защиты от DDoS-атак // Программные продукты и системы. 2019. Т. 32. № 3. С. 419–432. DOI: 10.15827/0236-235X.127.419-432.

For citation

Palchevsky E.V., Khristodulo O.I. Developing a self-learning method for a spiking neural network to protect against DDoS attacks. *Software & Systems*. 2019, vol. 32, no. 3, pp. 419–432 (in Russ.). DOI: 10.15827/0236-235X.127.419-432.