УДК 004.032.26+004.421.5

# ИСПОЛЬЗОВАНИЕ СИММЕТРИЧНОЙ НЕЙРОННОЙ СЕТИ ДЛЯ ГЕНЕРАЦИИ ПСЕВДОСЛУЧАЙНЫХ ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

*Рафеев П. Ю., студент гр.951008, Болтак С. В., ассистент*

*Белорусский государственный университет информатики и радиоэлектроники*
*г. Минск, Республика Беларусь*

*Болтак С.В. – ассистент*

**Аннотация.** В работе исследуется возможность создания генератора псевдослучайных числовых последовательностей (ПЧП) для потоковых криптосистем на базе симметричной нейронной сети. Описываются необходимые параметры нейронной сети, производится сравнение с другими генераторами ПЧП и делаются выводы о преимуществах и недостатках такого генератора.

**Ключевые слова.** Симметричная нейронная сеть, потоковое шифрование, потоковая криптосистема, генератор ключа, псевдослучайная числовая последовательность.

В современной криптографии большой популярностью пользуются т. н. «потоковые криптосистемы». Основная их идея заключается в шифровании исходного текста *M* с помощью побитовой операции XOR с криптографическим ключом *K*, длина которого равняется длине текста. Однако, поскольку обмен ключами большого размера зачастую невозможен, на практике для формирования ключевого потока используют генераторы псевдослучайной последовательности (LFSR, A5, MUGI, RC4 и др.) [1, с. 96—114]. Каждый из них обладает своими достоинствами и недостатками, что обуславливает их применимость лишь в ограниченных рамках. Так, генераторы LFSR и A5 легко реализуются аппаратно и не требуют больших затрат памяти, но из-за побитовой линейной генерации работают медленнее, обладают меньшим периодом и более предсказуемы, чем RC4 (работающий на основе подстановочной таблицы S) или MUGI (реализующий смешанную схему). Проблемой же использования подстановочной таблицы S является необходимость её предварительного заполнения и постоянного хранения в памяти, что значительно усложняет аппаратную реализацию.

Для решения описанных проблем различных генераторов в данной работе предлагается использование симметричной нейронной сети. Симметричность необходима для возможности подачи значений выходных нейронов на вход при дальнейшей генерации псевдослучайной числовой последовательности, а также распараллеливания вычислений. За один цикл работы нейросеть генерирует *k* чисел размерностью *n* бит (где *k* — количество входных/выходных нейронов). Чем больше *n* и *k*, тем больше длина генерируемой за один цикл последовательности и максимальный её период. Однако также бóльшими станут время вычисления цикла и обучения нейросети. При использовании же нейросети без предварительного обучения (со случайными весами нейронных связей) период последовательности может оказаться меньше максимального (равного $2^{n*k}$).

В созданной для данной работы нейронной сети $n$ = 32 (значения типа int) и $k$ = 4, что достаточно для генерации надёжного ключа размером до $2^{85}$ ТБ. Взлом же такого генератора, даже при известных *2*n*k* бит ключа, остаётся вычислительно трудной задачей. Количество промежуточных слоёв = 1 для снижения времени вычисления цикла. В качестве функции активации нейронов используется выгнутая тождественная функция [2, с.136], представленная формулой (1):

$$\mathrm{f(x)} = \left(\frac{\sqrt{\mathrm{x}^2+1}-1}{2} + \mathrm{x}\right) \bmod 2^{\mathrm{n}}, \tag{1}$$

где $x = \sum_{i=1}^{k} v_i \cdot f_i$ — сумма произведений выходных значений нейронов предыдущего слоя и весов их связей с активируемым нейроном (входное значение активируемого нейрона).

Такая функция активации обеспечивает нелинейность генерируемой выходной последовательности (поскольку сама функция не линейна), равномерное распределение значений в максимально широком диапазоне (благодаря своей непрерывности и положительной производной на всей области определения), более высокую скорость обучения нейросети (по причине монотонности производной, гарантирующей выпуклость поверхности ошибок), а также возможность аппаратной реализации нейронов (из-за существования радиоэлементов с такой же ВАХ).

Для обучения нейронной сети использовался генетический алгоритм, функцией приспособленности в котором выступал максимальный период выходной последовательности, генерируемой сетью с выбранными весами нейронных связей. Иные алгоритмы обучения, такие как градиентный спуск или обратное распространение ошибки в данной задаче плохо применимы, поскольку каждое вычисление функции ошибки требует выполнять генерацию выходной

последовательности до достижения повторения.

В данном алгоритме весовые коэффициенты всех нейросетей первого поколения выбирались случайно, а в последующих поколениях вычислялись на основании двух нейросетей с наибольшими периодами в предыдущем поколении по формуле (2):

$$v_{i+1} = (v_{max})_i + \left((v_{pred})_i - (v_{max})_i\right) \cdot \mathrm{rand}, \quad (2)$$

где *v* — вес связи, индексы i, i+1 — номера поколений, индексы max и pred означают нейросети с максимальным и предмаксимальным периодами соответственно, а rand — случайное значение в диапазоне от 0 до 1.

Кроме того, в одну нейросеть следующего поколения значения весов связей переносились из нейросети с максимальным периодом без изменений во избежание возможного регресса при обучении.

В каждом поколении параллельно вычислялся период повторения для 50 нейросетей (число выбрано оптимальным для компьютера автора). Условием окончания обучения являлось равенство периодов всех нейросетей в поколении, поскольку при его достижении фомула (2) даёт $v_{i+1} = v_i$. В данной работе до достижения условия окончания обучения прошло 38 поколений, информация о которых представлена в таблице 1.

Таблица 1 — Время достижения и наибольшие периоды каждого поколения нейросетей.

| Поколение нейросетей | Время (мин) | Период ($10^{21}$ бит) | |
|---|---|---|---|
| | | максимальный | предмаксимальный |
| 1 | 9,914547619 | 1,205762172 | 0,998113197 |
| 2 | 15,46908333 | 25,94993964 | 1,205762172 |
| 3 | 23,89047619 | 25,94993964 | 8,210723091 |
| 4 | 33,44666667 | 25,94993964 | 5,737463848 |
| 5 | 42,6445 | 27,46440986 | 25,94993964 |
| 6 | 57,57604762 | 31,54231382 | 27,46440986 |
| 7 | 68,56566667 | 31,54231382 | 29,86093363 |
| 8 | 74,47855952 | 60,5082547 | 31,54231382 |
| 9 | 82,36241667 | 60,5082547 | 25,94993964 |
| 10 | 87,02105952 | 60,5082547 | 37,99330663 |
| 11 | 91,50052381 | 77,42654066 | 60,5082547 |
| 12 | 96,57725 | 77,42654066 | 63,35434483 |
| 13 | 99,86219048 | 77,42654066 | 41,0412532 |
| 14 | 103,4457619 | 77,42654066 | 44,2690102 |
| 15 | 106,7307024 | 77,42654066 | 38,99020063 |
| 16 | 110,2545476 | 77,42654066 | 37,99330663 |
| 17 | 113,8978452 | 77,42654066 | 53,80979485 |
| 18 | 117,2425119 | 77,42654066 | 74,11562262 |
| 19 | 120,2885476 | 82,60307038 | 77,42654066 |
| 20 | 123,3943095 | 82,60307038 | 79,12418529 |
| 21 | 126,4403452 | 82,60307038 | 79,12418529 |
| 22 | 129,486381 | 86,19288672 | 82,60307038 |
| 23 | 144,0595714 | 88,03234242 | 86,19288672 |
| 24 | 154,6908333 | 88,03234242 | 72,49944514 |
| 25 | 159,7078333 | 88,03234242 | 79,12418529 |
| 26 | 163,4705833 | 88,03234242 | 84,3824096 |
| 27 | 167,7111429 | 101,7402099 | 88,03234242 |
| 28 | 171,4141667 | 101,7402099 | 93,72495476 |
| 29 | 174,6991071 | 101,7402099 | 91,79675321 |
| 30 | 177,7451429 | 101,7402099 | 99,68928814 |
| 31 | 181,1495357 | 112,4672973 | 101,7402099 |
| 32 | 184,2552976 | 112,4672973 | 108,0801462 |
| 33 | 187,7791429 | 126,4394341 | 112,4672973 |
| 34 | 191,601619 | 126,4394341 | 124,0244369 |
| 35 | 196,7977976 | 133,8928274 | 126,4394341 |
| 36 | 208,6235833 | 141,6709874 | 133,8928274 |
| 37 | 222,300881 | 141,6709874 | 141,6709874 |
| 38 | 237,9491429 | 141,6709874 | 141,6709874 |

По данным таблицы 1 построены графики зависимости максимального и предмаксимального периодов в каждом поколении от времени достижения поколения (кривые обучения), изображённые на рисунке 1:

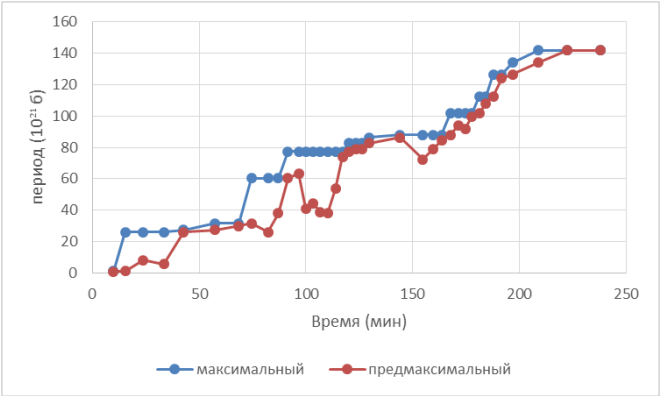


Рисунок 1 – Графики зависимости двух лучших периодов в поколении от времени обучения по данным таблицы 1

Из кривых обучения видно, что копирование весовых коэффициентов лучшей нейронной сети предыдущего поколения в одну из нейронных сетей следующего поколения действительно необходимо и позволяет избежать регресса при обучении, вызванного негладкостью поверхности ошибок (которая отражается на графике зависимости предмаксимального периода от времени как ряд локальных минимумов). Также графики и таблица показывают преимущество обученной нейросети над необученной в максимальном периоде выходной последовательности (120-кратное преимущество), в большинстве ситуаций оправдывающее необходимость тратить несколько часов на предварительное обучение.

Весовые коэффициенты, полученные после 38-ми поколений генетического алгоритма и используемые в нейросети, представлены в таблице 2.

Таблица 2 — Используемые в нейронной сети весовые коэффициенты.

| Внутренний слой | | | | | Выходной слой | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $№_{вх.}/№_{вых.}$ | 1 | 2 | 3 | 4 | $№_{вх.}/№_{вых.}$ | 1 | 2 | 3 | 4 |
| 1 | 478.9 | 9823.1 | 10056.4 | 3609.4 | 1 | 7896.5 | 7583.4 | 3794.1 | 749.4 |
| 2 | 2274.0 | 5824.8 | 6673.5 | 10598.3 | 2 | 3713.6 | 5732.4 | 5374.5 | 3563.0 |
| 3 | 358.1 | 4962.3 | 5829.1 | 10037.1 | 3 | 4782.9 | 3210.6 | 6473.7 | 9683.6 |
| 4 | 4891.4 | 708.3 | 5032.5 | 9028.5 | 4 | 146.0 | 8334.1 | 7824.7 | 208.7 |

После завершения создания и обучения нейросети необходимо было проверить, является ли распределение цифр в генерируемой выходной последовательности равномерным, для чего была сгенерирована выходная последовательность из 48 миллионов цифр, подсчитано количество цифр каждого вида в ней и вычислены их относительные частоты появлений. Результаты представлены в таблице 3.

Таблица 3 — Количества и относительные частоты цифр в выходной последовательности.

| Цифра | 0 | 1 | 2 | 3 | 4 | - |
|---|---|---|---|---|---|---|
| Кол-во | 4 799 825 | 4 800 594 | 4 800 338 | 4 799 796 | 4 801 443 | - |
| Частота, % | 10,000 | 10,001 | 10,001 | 10,000 | 10,003 | - |
| Цифра | 5 | 6 | 7 | 8 | 9 | В сумме |
| Кол-во | 4 799 882 | 4 798 103 | 4 797 842 | 4 799 595 | 4 802 582 | 48 000 000 |
| Частота, % | 10,000 | 9,996 | 9,996 | 9,999 | 10,005 | 100,000 |

Для оценки близости распределения к равномерному был вычислен критерий согласия Пирсона [3, с. 138—140] по формуле (3):

$$\chi^2 = \sum \frac{(n_i - n_i')^2}{n_i'}, \quad (3)$$

где $n_i$ и $n_i'$ — экспериментальные и теоретические частоты соответственно.

Полученное значение критерия $\chi^2 = 7{,}69*10^{-8}$ при числе степеней свободы k = S - 2 = 8 (где S — количество видов цифр) означает, что вероятность не равномерного распределения цифр порядка $10^{-30}$ %, и свидетельствует о равномерном распределении цифр в генерируемой ПЧП — необходимом для большинства псевдослучайных генераторов качестве.

Далее было проведено сравнение нейросети по скорости генерации псевдослучайной числовой последовательности с генераторами LFSR и RC4. Результаты приведены в таблице 4 (последняя строка таблицы представляет собой сравнение теоретических сложностей).

Таблица 4 — Сравнение времени (в секундах) генерации ПЧП различными генераторами.

| Длина,МБ/Генератор | Нейросеть | LFSR | RC4 |
|---|---|---|---|
| 8 | 0,79 | 5,29 | 0,76 |
| 16 | 1,38 | 10,41 | 1,40 |
| 32 | 2,48 | 20,71 | 2,47 |
| 64 | 4,74 | 42,45 | 4,48 |
| 128 | 8,66 | 76,39 | 8,35 |
| 192 | 13,13 | 116,09 | 12,45 |
| 256 | 17,21 | 153,4 | 16,51 |
| *N* | O(2**k***N*/*n*) | O(2**N*) | O(7**N*/*n*) |

По данным таблицы 4 для большей наглядности сравнения построены графики зависимости времени генерации от длины ПЧП, изображённые на рисунке 2:

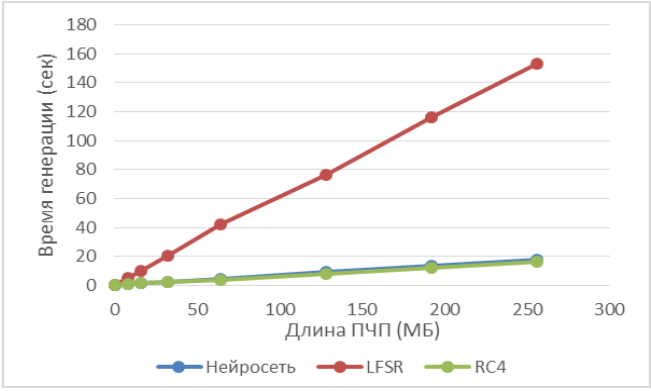


Рисунок 2 – Графики зависимости времени генерации от длины ПЧП по данным таблицы 2

Как видно из результатов сравнения, нейросеть смогла достичь такой же скорости, как и генератор RC4, при этом требуя гораздо меньших затрат памяти (2**k* против 256), обладая большой гибкостью (за счёт возможности переобучить её, не меняя исходный код) и возможностью аппаратной реализации на базе современных нейронных процессоров с использованием радиодеталей, ВАХ которых аппроксимируется функцией активации [4]. Основным её недостатком можно считать необходимость длительного предварительного обучения, что, однако, можно решить путём переноса весовых коэффициентов уже обученной нейросети в новую. Таким образом, применение симметричных нейронных сетей может послужить универсальным решением для генерации ПЧП, обеспечивая быстрое, ресурсоёмкое и криптографически стойкое потоковое шифрование.

***Список использованных источников:***

*1. Ярмолик, В. Н. Элементы теории информации: Конспект лекций по одноименной дисциплине для студентов направления специальности "Программное обеспечение информационных технологий" дневной и заочной формы обучения / В. Н. Ярмолик — Минск : БГУИР, 2015. — 141с.*

*2. Чару Аггарвал. Нейронные сети и глубокое обучение. Учебный курс. Пер. с анг. А. Г. Гузикевич / Чару Аггарвал. — М. : Вильямс, 2020. — 752 с.*

*3. Аксенчик, А. В. Теория вероятностей и математическая статистика : учебно - метод. пособие / А. В. Аксенчик. – Минск : БГУИР, 2011. – 184 с.*

*4. Гриняев С. / Нейронные процессоры Intel [Электронный ресурс] // Компьютерра. 2001. — № 38. — Режим доступа: https://old.computerra.ru/198014/— Дата доступа: 03.04.2021.*

UDC 004.032.26+004.421.5

# SYMMETRIC NEURAL NETWORK AS PSEUDORANDOM NUMERICAL SEQUENCES GENERATOR

*Rafeyev P. Y, Boltak S. V.*

*Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

*Boltak S. V. – assistant*

**Annotation.** This article researches the possibility of creating a pseudorandom number sequence generator (PNSG) for stream cryptosystems based on a symmetric neural network. The necessary parameters of the neural network are described, compared with other PNSG generators, and the conclusions about the advantages and disadvantages of such a generator are made.

**Keywords.** Symmetric neural network, stream encryption, stream cryptosystem, key generator, pseudorandom numeric sequence.