УДК 004.67

ПРОГРАММНЫЙ МОДУЛЬ ПОДДЕРЖКИ АЛГЕБРЫ ГИПЕРРАЗМЕРНЫХ ВЕКТОРОВ

Д. А. Трокоз, к.т.н., доцент, проректор по научной работе ФГБОУ ВО «Пензенский государственный технологический университет», Пенза, Россия;
orcid.org/0000-0003-0989-3928, e-mail: trokoz@penzgtu.ru

Предложено архитектурное решение для программного модуля поддержки алгебры гиперразмерных двоичных векторов. Данный модуль позволяет выполнять преобразование значений параметров целевой задачи в гиперразмерное представление, которое более эффективно в ряде математических моделей в сравнении с классическим позиционно-зависимым представлением данных. В основе алгоритма такого преобразования лежит метод инвариантного представления данных, использующий алгебру гиперразмерных двоичных векторов, который позволяет выполнить позиционно-независимое кодирование произвольной структуры данных.

Модуль использует сервисно-ориентированный подход, что существенно упрощает его применение за счет унификации протокола взаимодействия модуля с внешними системами, который подробно описан в работе. Кроме того, поскольку в алгоритме преобразований данных в гиперразмерное представление интенсивно применяются векторные операции, то предлагается вместо универсального процессора в качестве основного вычислительного устройства модуля использовать графический ускоритель, что позволит увеличить выполнение таких операций в десятки раз.

Ключевые слова: *программный модуль, временной ряд, алгебра гиперразмерных векторов, сервисно-ориентированный подход, позиционно-независимое кодирование, инвариантное представление, широкая нейронная сеть, параметрическая модель.*

DOI: 10.21667/1995-4565-2021-75-53-62

Введение

Математический аппарат гиперразмерных двоичных векторов [1] широко используется во многих эвристических алгоритмах и методах анализа данных [2-3]. Данная работа посвящена проектированию программного модуля поддержки алгебры гиперразмерных двоичных векторов. Создание такого модуля позволит упростить реализацию многих эвристических методов, делегировав ему задачу преобразования данных в гиперразмерное представление [4]. В проектируемом модуле предлагается реализовать метод инвариантного представления данных в нейронных сетях с использованием алгебры гиперразмерных двоичных векторов [5], который позволяет выполнить позиционно-независимое кодирование произвольной структуры данных [6].

Модуль предназначен для преобразования значений параметров математической модели в гиперразмерное представление. При этом должна быть реализована поддержка различных параметрических моделей для одного набора данных, что обеспечит необходимую гибкость при реализации широкого класса эвристических методов. В процессе проектирования программного модуля поддержки алгебры гиперразмерных векторов необходимо определить набор функций, который он должен поддерживать, описать алгоритм работы для каждой функции, а также протокол взаимодействия с этим модулем. При реализации модуля предлагается использовать архитектурный стиль REST [7-8] и формат JSON [9] для взаимодействия с внешними системами.

Описание алгоритмов работы модуля

Прежде чем определить набор функций, выполнение которых должен обеспечивать модуль поддержки алгебры гиперразмерных векторов, нужно учесть ряд важных факторов.

Преобразование данных в гиперразмерное представление является достаточно ресурсоемкой операцией в части как временных, так и объемных затрат (расход оперативной памяти), поэтому необходимо предусмотреть возможность многократного частичного использования результатов предыдущих вычислений для снижения потребления вычислительных ресурсов. Для этого крупные функции, которые бы позволяли сразу преобразовать массив наборов значений параметров в массив гиперразмерных векторов, нужно декомпозировать [10]. Предлагается отдельно преобразовать каждое значение каждого параметра в гиперразмерный вектор, а затем в зависимости от параметрической модели в результирующий гиперразмерный вектор включать только необходимые параметры. Это позволит только существенно снизить объем вычислений. Однако, это ведет к другой проблеме – существенному росту объема передаваемых данных [11], за счет передачи данных промежуточных вычислений. Для решения этой проблемы предлагается использовать методологию объектно-ориентиро-ванного проектирования [12-14]. В этом случае данные для преобразования интерпретируются как объект, который создается и хранится на стороне модуля поддержки алгебры гиперразмерных векторов, а

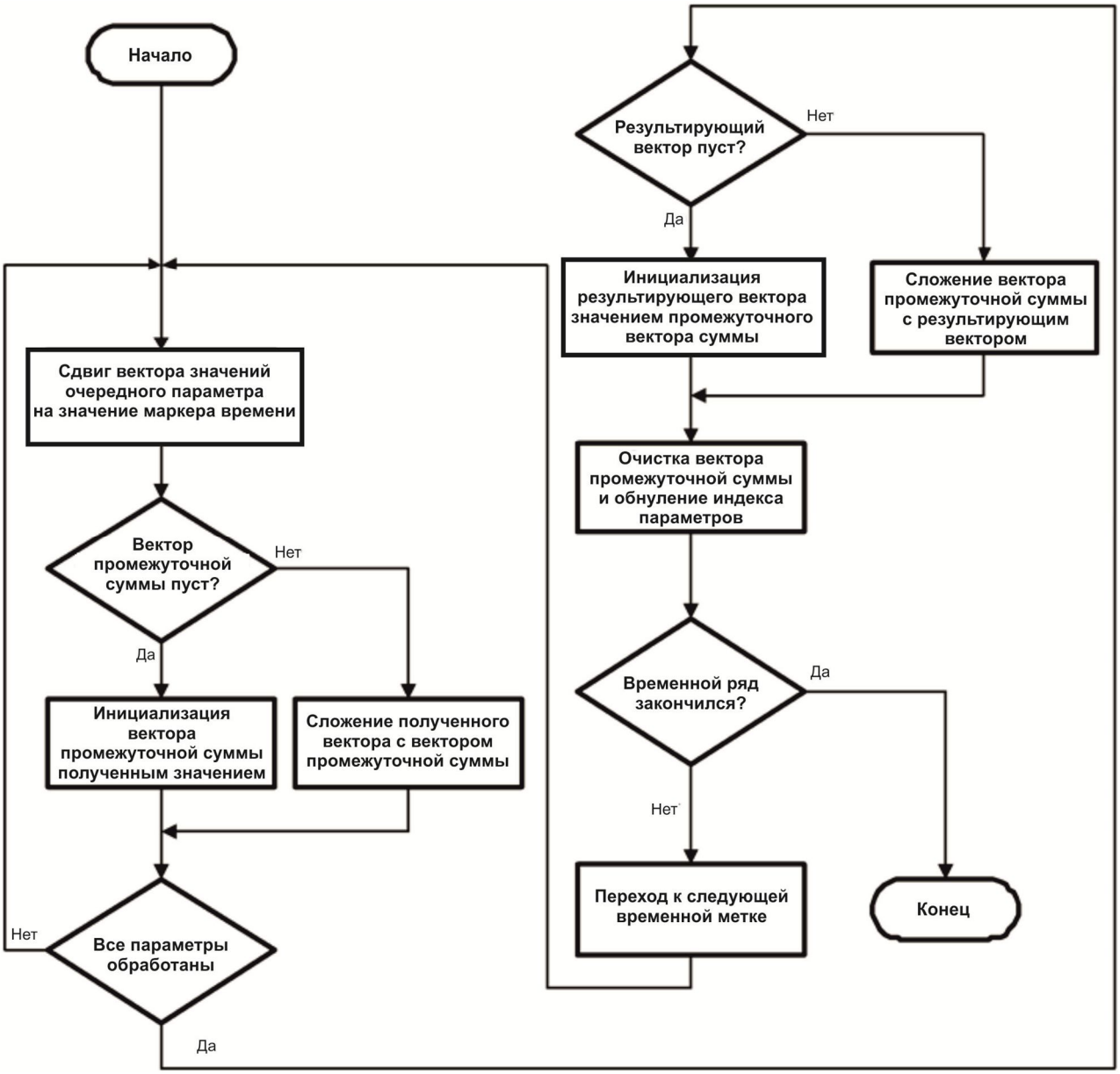

Рисунок 1 – Алгоритм формирования результирующего вектора при использовании временных рядов
Figure 1 – Forming the resulting vector when using time series

внешняя система вызывает методы для данного объекта, передавая лишь метаданные для преобразования [15]. С учетом сказанного предлагается следующий набор запросов и команд, которые могут быть переданы модулю поддержки алгебры гиперразмерных векторов:

1. Создание нового пустого набора данных по заданным метаданным, описывающим его структуру. В результате создается экземпляр структуры данных, который способен хранить гиперразмерные векторы для каждого значения описанных параметров. Идентификатор созданного экземпляра структуры возвращается внешней системе для вызова методов этого объекта в дальнейшем.

2. Добавление нового набора значений параметров для заданного экземпляра структуры данных. При выполнении этого метода происходит преобразование переданных значений параметров в гиперразмерное представление и сохранение полученных гиперразмерных векторов, а не исходных значений параметров.

3. Формирование массива результирующих гиперразмерных векторов для заданного экземпляра структуры данных. В запросе передается параметрическая модель, которая представляет собой массив номеров параметров, значения которых необходимо добавить в результирующий гиперразмерный вектор формируемого массива. Алгоритм формирования результирующего вектора для случая с временными рядами представлен на рисунке 2. Для случая, когда временные ряды не используются, алгоритм использовать только одним набор параметров.

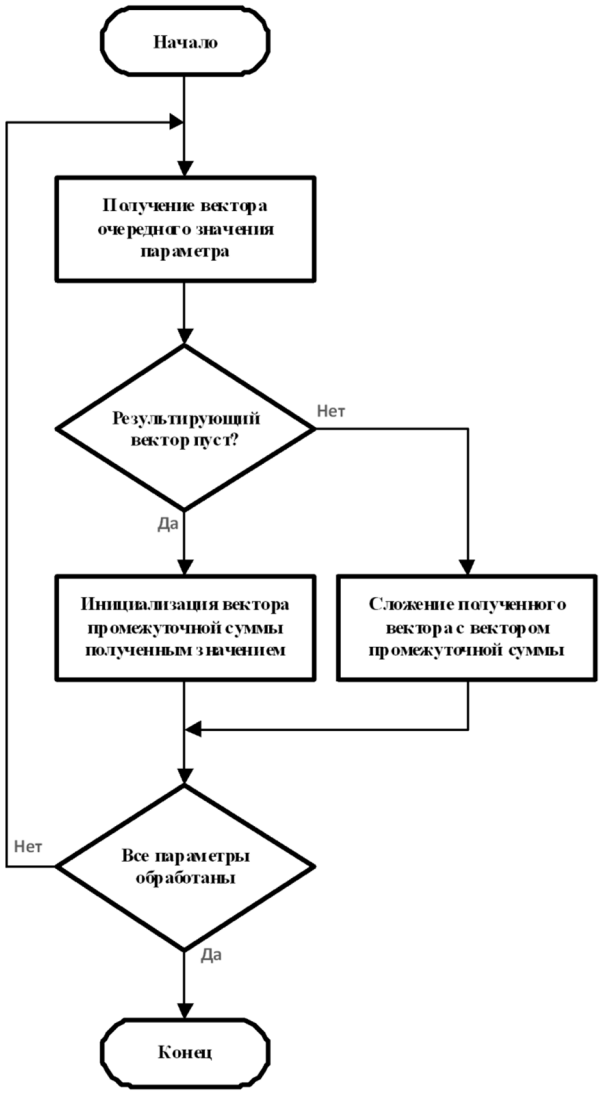

Рисунок 2 – Алгоритм формирования результирующего вектора без использования временных рядов
Figure 2 – Algorithm for generating the resulting vector without using time series

4. Удаление заданного экземпляра структуры данных. Выполняется, когда работа со структурой завершена и требуется освободить занятую оперативную память.

Описание протокола взаимодействия модуля с внешними системами

Модуль поддержки алгебры гиперразмерных векторов взаимодействует только с внешними системами, которые реализуют основной эвристический алгоритм, использующий алгебру гиперразмерных двоичных векторов. Рассмотрим диаграмму последовательности, которая бы показывала такое взаимодействие.

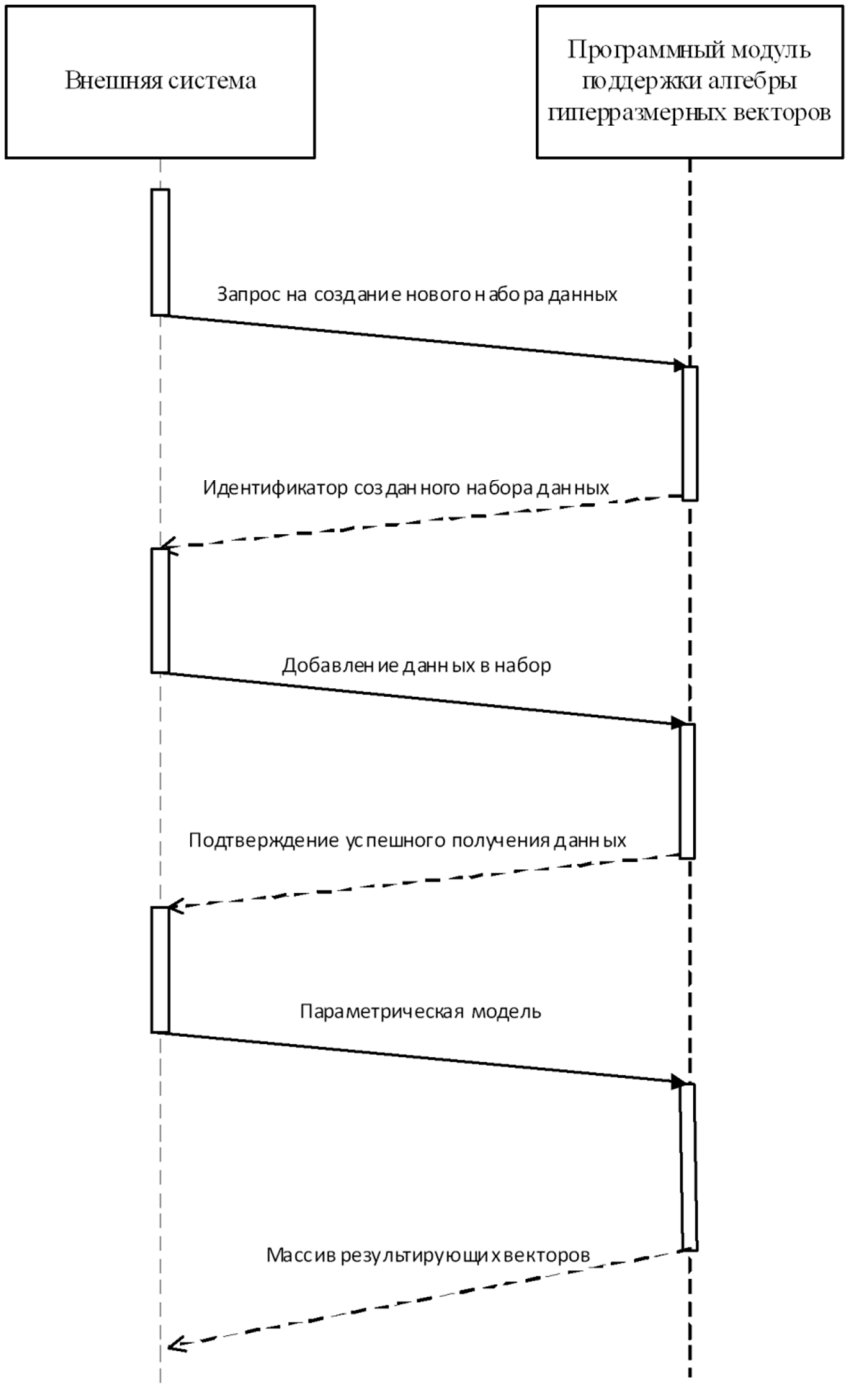

Рисунок 3 – Обмен сообщениями внешней системы с модулем поддержки алгебры гиперразмерных векторов
Figure 3 – External system messaging with hyperdimensional vector algebra support module

Рассмотрим подробно структуру сообщений каждого запроса и ответа:

1. Запрос на создание нового набора данных.

Внешняя система передает модулю поддержки алгебры гиперразмерных векторов запрос в виде JSON сообщения на создание нового набора данных. Запрос содержит метаинформацию, необходимую для описания структуры набора данных. Она содержит массив с количеством бит для каждого параметра, а также признак того, используется ли единичный набор параметров или массив наборов параметров, представляющий собой временной ряд. С учетом сказанного предлагается следующая структура JSON запроса на создание нового набора данных:

```
{
   "bitlen": [p1, p2, …, pm],
   "isArray": isArray
}
```

где

– p1, p2, …, pm – массив чисел, содержащих количество бит соответствующего параметра;

– isArray – булево значение, указывающее, предполагается ли дублирование структуры в виде временного ряда.

2. Идентификатор созданного набора данных.

Модуль поддержки алгебры гиперразмерных векторов передает ответ на запрос создания структуры данных внешней системе. Ответ представляет собой JSON структуру с одним полем id, которое содержит уникальный идентификатор структуры данных GUID длиной 128 бит. Указанный JSON с ответом имеет следующий вид:

```
{
 "id": guid
}
```

где guid – уникальный 128-битный идентификатор структуры данных. Данный идентификатор используется в дальнейшем в других запросах для выполнения различных действий с созданной структурой данных. Возвращаемый код состояния HTTP 200 в случае успешного выполнения запроса. В противном случае будут возвращены код состояния HTTP 400 и сообщение об ошибке [16, 17].

3. Добавление данных в набор.

Внешняя система передает модулю поддержки алгебры гиперразмерных векторов запрос в виде JSON сообщения, содержащий данные, которые необходимо добавить в указанный набор. Запрос должен содержать уникальный идентификатор созданной ранее структуры данных и добавляемые данные [18]. В зависимости от того, был ли при создании структуры установлен флаг, указывающий, что структура представляет собой временной ряд, возможны два варианта запроса.

С использованием временных рядов [19]:

```
{
 "id": "guid",
 "data": [
  [
   {
    "time": t1,
    "data": [p1/1, p1/2, ..., p1/m]
   },
   ...,
   {
    "time": t1,
    "data": [pi/1, pi/2, ..., pi/m]
   },
```

```
        ...,
        {
          "time": tk,
          "data": [pk/1, pk/2, ..., pk/m]
        }
      ],
      ...
    ]
  }
```

где

– guid – уникальный 128-битный идентификатор структуры, для которой добавляются данные;

– pi/1, pi/2, ..., pi/m – массив значений параметров для i-й временной метки;

– ti – значение i-й временной метки.

Без использования временных рядов:

```
{
  "id": "guid",
  "data": [
    [p1, p2, ..., pm],
    ...
  ]
}
```

где

– guid – уникальный 128-битный идентификатор структуры, для которой добавляются данные;

– p1, p2, ..., pm – массив значений параметров.

В ходе выполнения запроса каждый из добавляемых параметров преобразуется в гиперрамзерный двоичный вектор и сохраняется в таком виде. Это позволяет выполнять такое преобразование однократно при добавлении.

4. Подтверждение успешного получения данных.

Модуль поддержки алгебры гиперразмерных векторов передает ответ на запрос добавления внешней системе. Тело запроса пусто, а возвращаемый код состояния HTTP 200 в случае успешного выполнения запроса. В противном случае будут возвращены код состояния HTTP 400 и сообщение об ошибке.

5. Параметрическая модель.

Внешняя система передает модулю поддержки алгебры гиперразмерных векторов запрос в виде JSON сообщения на получение массива гиперразмерных двоичных векторов, представляющих собой набор данных, в соответствии с переданной параметрической моделью [20]. Параметрическая модель, переданная в запросе, указывает, какие параметры нужно включить в результирующие гиперразмерные векторы. Запрос содержит уникальный идентификатор созданной ранее структуры данных, для которой были добавлены данные, и параметрическую модель, представляющую собой массив номеров параметров. С учетом сказанного предлагается следующая структура JSON для данного запроса:

```
{
  "id": "guid",
  "params": [pn1, pn2, ..., pnk]
}
```

где

– guid – уникальный 128-битный идентификатор структуры, для которой добавляются данные;

– pn1, pn2, ..., pnk – массив номеров параметров.

В ходе выполнения запроса параметры, указанные в параметрической модели, объединяются в один гиперразмерный вектор для каждого набора данных или временного ряда, формируя массив гиперразмерных двоичных векторов, который будет представлять собой в дальнейшем входные или выходные данные для нейронной сети. Метод представления данных в широких нейронных сетях с использованием алгебры гиперразмерных двоичных векторов, который применяется для данного преобразования подробно рассмотрен во второй главе.

6. Массив результирующих векторов.

Модуль поддержки алгебры гиперразмерных векторов передает ответ на запрос формирования гиперразмерного представления данных для заданной параметрической модели внешней системе. Ответ содержит массив гиперразмерных двоичных векторов, каждый из которых представляет собой один набор данных или временной ряд, переданные ранее. С учетом сказанного предлагается следующая структура JSON ответа:

[hdv1, hdv2, ..., hdvm]

где hdv1, hdv2, ..., hdvm – массив результирующих гиперразмерных векторов.

Предложенный механизм организации взаимодействия модуля поддержки алгебры гиперразмерных векторов с внешней системой позволяет существенно снизить объем передаваемых данных, поскольку избавляет от многократной передачи одних и тех же данных с разными параметрическими моделями. А также снижает вычислительную нагрузку на аппаратную часть, потому что первичное преобразование параметров в гиперразмерное представление производится однократно в момент передачи данных, при этом сохраняется именно сформированное гиперразмерное представление параметров, а не исходные данные. Для формирования данных для очередной параметрической модели достаточно передать только саму модель и идентификатор структуры данных, гиперразмерное представление загруженных ранее наборов данных которой требуется сформировать.

Заключение

Реализация программного модуля поддержки алгебры гиперразмерных двоичных векторов, представленная в работе, позволяет эффективно организовать преобразование значений параметров математических моделей в гиперразмерное представление. Сервисно-ориентированный подход, использованный при проектировании модуля, обеспечивает необходимую гибкость его использования внешними системами, за унификации протокола взаимодействия.

Поскольку гиперразмерные преобразования, с одной стороны, требуют большого количества вычислительных ресурсов, а с другой – хорошо поддаются распараллеливанию, то в качестве аппаратной вычислительной платформы целесообразно выбрать не классический универсальный микропроцессор, а графический ускоритель. Он позволит ускорить выполнение векторных операций в десятки раз по сравнению с универсальным процессором аналогичной стоимости. А именно векторные операции составляют более 99 % всех вычислений, выполняемых модулем.

Предложенное архитектурное решение модуля поддержки алгебры гиперразмерных двоичных векторов может использоваться при программной реализации широкого класса математических моделей, использующих гиперразмерное представление данных в вычислениях, позволяя абстрагироваться от преобразования значений параметров целевой задачи в гиперразмерный вид.

Библиографический список

1. **Пащенко Д. В. и др.** Использование алгебры гиперразмерных векторов для эвристического представления данных при обучении широких нейронных сетей // Проблемы управления и моделирования в сложных системах. 2019. С. 301-305.

2. **Минкина Т. В., Шендриков Н. В.** Эвристическая модель, основы имитирования реальных процессов и явлений // Культура и общество: история и современность. 2013. С. 111-115.

3. **Ильясов И. И.** Система эвристических приемов решения задач. М.: Изд-во Российского открытого ун-та. 1992. 140 с.

4. **Чернов К. В.** Системнология безопасности. 2011. 195 с.

5. **Kanerva P.** Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors // Cognitive Computation. 2009, vol 1, no. 2, pp. 139-159.

6. **Pashchenko D., Trokoz D., Martyshkin A. and Sinev M.** Using Algebra of Hyper-Dimensional Vectors for Heuristic Representation of Data While Training Wide Neural Networks // 2019 XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP), Samara, Russia, 2019, pp. 168-171.

7. **Чеглаков А. Л.** Композиция WEB-сервисов на основе архитектуры REST // Инновационная наука. 2016. №. 12-2. С. 118-120.

8. **Суринов А. С., Колчин М. А.** Исследование и разработка веб-интерфейсов прикладного программирования на основе архитектурного стиля REST // Альманах научных работ молодых ученых Университета ИТМО. 2017. С. 298-300.

9. **Crockford D. Json** // ECMA International. 2012.

10. **Золотов В. А., Семенов В. А.** Исследование и развитие метода декомпозиции для анализа больших пространственных данных // Труды Института системного программирования РАН. 2013. Т. 25. С. 131-166.

11. **Исаев Е. А., Корнилов В. В.** Проблема обработки и хранения больших объемов научных данных и подходы к ее решению // Математическая биология и биоинформатика. 2013. Т. 8. №. 1. С. 49-65.

12. **Терехов А. Н. и др.** Объектно-ориентированная методология разработки информационных систем и систем реального времени // Объектноориентированное визуальное моделирование. 1999. С. 4-20.

13. **Силич В. А., Силич М. П.** Проектирование сложной системы на основе объектно-ориентированного подхода // Известия Томского политехнического университета. Инжиниринг георесурсов. 2003. Т. 306. №. 2. С. 99-103.

14. **Морозова Е. В.** Проблемы введения объектно-ориентированного программирования // Прогрессивные технологии в обучении и производстве. Материалы Всероссийской конференции: Межвузовский сборник, г. Камышин. 2004. С. 154-156.

15. **Радченко Г. И.** Распределенные вычислительные системы: учеб. пособие / Г. И. Радченко. Челябинск: Фотохудожник. 2012. 184 с.

16. **Григорьев А. С.** Применение современных веб-технологий в разработке кроссплатформенных программных продуктов на примере создания онлайн-приложения: дис. – Южно-Уральский государственный университет. 2019.

17. **Титаев А. А.** Информационные сети и телекоммуникации. 2017. 98 с.

18. **Левчук Е.** Технологии организации, хранения и обработки данных. 2017. 241 с.

19. **Клячкин В. Н., Кувайскова Ю. Е., Бубырь Д. С.** Прогнозирование состояния объекта с использованием систем временных рядов // Радиотехника. 2015. №. 6. С. 45-47.

20. **Прошин И. А., Прошин Д. И., Прошина Р. Д.** Структурно-параметрический синтез математических моделей объектов исследования по экспериментальным данным // Вестник Астраханского государственного технического университета. Серия: Морская техника и технология. 2009. №. 1. С. 110-115.

UDC 004.67

SOFTWARE MODULE FOR SUPPORTING THE ALGEBRA OF HYPERDIMENSIONAL VECTORS

D. A. Trokoz, PhD (technical sciences), Associate Professor, Vice-Rector for Scientific Work, Penza State Technological University, Penza, Russia;
orcid.org/0000-0003-0989-3928, e-mail: trokoz@penzgtu.ru

The paper proposes an architectural solution for a software module to support the algebra of hyperdimensional binary vectors. This module allows converting the values of target problem parameters into a hyperdimensional representation, which is more efficient in a number of mathematical models in comparison with classical position-dependent representation of data. The algorithm of such a transformation is based on the method of invariant data representation, using the algebra of hyperdimensional binary vectors, which allows performing position-independent encoding of arbitrary data structure.

The module uses a service-oriented approach, which greatly simplifies its use, due to the unification of the protocol for interaction of the module with external systems, which is described in detail in the work. In addition, since vector operations are intensively used in the algorithm for transforming data into hyperdimensional representation, the authors have proposed to use graphics accelerator instead of general-purpose processor as the main computing device of the module, which will tenfold increase the execution of such operations.

Key words: *software module, time series, hyperdimensional vector algebra, service-oriented approach, position-independent coding, invariant representation, wide neural network, parametric model.*

DOI: 10.21667/1995-4565-2021-75-53-62

References

1. **Paschenko D. V i dr.** Ispol'zovanie algebry giperrazmernyh vektorov dlya evristiche-skogo predstavleniya dannyh pri obuchenii shirokih nejronnyh setej. *Problemy upravleniya i modelirovaniya v slozhnyh sistemah*. 2019, pp. 301-305. (in Russian).

2. **Minkina T. V., SHendrikov N. V.** Evristicheskaya model', osnovy imitirovaniya real'nyh processov i yavlenij. *Kul'tura i obshchestvo: istoriya i sovremennost'*. 2013, pp. 111-115. (in Russian).

3. **Ilyasov I. I.** *Sistema evristicheskih priemov resheniya zadach* (The system of heuristic methods of solving problems). Moscow: Izd-vo Rossijskogo otkrytogo un-ta. 1992, 140 p. (in Russian).

4. **Chernov K. V.** *Sistemnologiya bezopasnosti* (Security systemology). 2011, 195 p. (in Russian).

5. **Kanerva P.** Hyperdimensional Computing: An Introduction to Computing in Distributed Representation with High-Dimensional Random Vectors. *Cognitive Computation*. 2009, vol 1, no. 2, pp. 139-159.

6. **Pashchenko D., Trokoz D., Martyshkin A. and Sinev M.** Using Algebra of Hyper-Dimensional Vectors for Heuristic Representation of Data While Training Wide Neural Networks. *2019 XXI International Conference Complex Systems: Control and Modeling Problems (CSCMP)*, Samara, Russia, 2019, pp. 168-171.

7. **Cheglakov A. L.** Kompoziciya WEB-servisov na osnove arhitektury REST. *Innovacionnaya nauka*. 2016, no. 12-2, pp. 118-120. (in Russian).

8. **Surinov A. S., Kolchin M. A.** Issledovanie i razrabotka veb-interfejsov prikladnogo programmirovaniya na osnove arhitekturnogo stilya REST. *Al'manah nauchnyh rabot molodyh uchenyh Universiteta ITMO*. 2017, pp. 298-300. (in Russian).

9. **Crockford D. Json**. *ECMA International*. 2012.

10. **Zolotov V. A., Semenov V. A.** Issledovanie i razvitie metoda dekompozicii dlya analiza bol'shih prostranstvennyh dannyh. *Trudy Instituta sistemnogo programmirovaniya RAN*. 2013, vol. 25, pp. 131-166. (in Russian).

11. **Isaev E. A., Kornilov V. V.** Problema obrabotki i hraneniya bol'shih ob"emov nauchnyh dannyh i podhody k ee resheniyu. *Matematicheskaya biologiya i bioinformatika*. 2013, vol. 8, no. 1, pp. 49-65. (in Russian).

12. **Terekhov A. N. et al**. Ob"ektno-orientirovannaya metodologiya razrabotki informacionnyh sistem i sistem real'nogo vremeni. *Ob"ektnoorientirovannoe vizual'noe modelirova-nie*. 1999, pp. 4-20. (in Russian).

13. **Silich V. A., Silich M. P.** Proektirovanie slozhnoj sistemy na osnove ob"ektno-orientirovannogo podhoda. *Izvestiya Tomskogo politekhnicheskogo universiteta. Inzhiniring georesursov*. 2003, vol. 306, no. 2, pp. 99-103. (in Russian).

14. **Morozova E. V.** Problemy vvedeniya ob"ektno-orientirovannogo programmirovaniya. *Progressivnye tekhnolo-gii v obuchenii i proizvodstve. Materialy Vserossijskoj konferencii: Mezhvuzovskij sbornik, g. Kamyshin*. 2004, pp. 154-156. (in Russian).

15. **Radchenko G. I.** *Raspredelennye vychislitel'nye sistemy:* ucheb. posobie / G. I. Radchenko. Chelyabinsk: Fotohudozhnik. 2012. 184 p. (in Russian).

16. **Grigoriev A. S.** *Primenenie sovremennyh veb-tekhnologij v razrabotke krossplatfor-mennyh programmnyh produktov na primere sozdaniya onlajn-prilozheniya:* dis. Yuzhno-Ural'skij gosudarstvennyj universitet, 2019. (in Russian).

17. **Titaev A. A**. *Informacionnye seti i telekommunikacii*. 2017, 98 p. (in Russian).

18. **Levchuk E**. T*ekhnologii organizacii, hraneniya i obrabotki dannyh*. 2017, 241 p. (in Russian).

19. **Klyachkin V. N., Kuvaiskova Yu. E., Bubyr D. S.** Prognozirovanie sostoyaniya ob"ekta s ispol'zovaniem sistem vremennyh ryadov. *Radiotekhnika.* 2015, no. 6, pp. 45-47. (in Russian).

20. **Proshin I. A., Proshin D. I., Proshina R. D**. Strukturno-parametricheskij sintez ma-tematicheskih modelej ob"ektov issledovaniya po eksperimental'nym dannym. *Vestnik Astrahanskogo gosudarstvennogo tekhnicheskogo universiteta. Seriya: Morskaya tekhnika i tekhnologiya*. 2009, no. 1, pp. 110-115. (in Russian).