

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И МАШИННОЕ ОБУЧЕНИЕ

УДК 004.89

**Исследование декомпозиции нейронной сети
в системе схемотехнического моделирования Proteus**

В. В. Бахтин¹, И. А. Подлесных², С. Ф. Тюрин³

¹Пермский национальный исследовательский политехнический университет; Пермь, Россия
e-mail: bakhtin_94@bk.ru; **ORCID:** 0000-0003-4873-7741, **AuthorID:** 882760

²Пермский национальный исследовательский политехнический университет; Пермь, Россия
e-mail: podlesnihwork@gmail.com; **ORCID:** 0000-0001-6260-393X, **AuthorID:** 1149281

³Пермский национальный исследовательский политехнический университет; Пермь, Россия
Пермский государственный национальный исследовательский университет; Пермь, Россия
e-mail: tyurinsergfeo@yandex.ru; **ORCID:** 0000-0002-5707-5404, **AuthorID:** 629677

Рассматривается разделение монолитной нейронной сети на блоки в рамках концепции туманных вычислений (Fog computing). Предполагается, что с учетом возможной реконфигурации реализация блоков выполняется на программируемой логике: ПЛИС (field-programmable gate array, FPGA, complex programmable logic device, CPLD), системах на кристалле (System-on-a-Chip, SoC) или системах в пакете (System-in-Package, SiP). В статье исследуется такая реализация в системе схемотехнического моделирования Proteus на примере микроконтроллеров ATmega32. Моделирование подтверждает эффективность разработанного метода декомпозиции. Исследования выполнены по гранту РФФИ 20-37-90036 (Метод синтеза устройств нейросетевого распознавания для реализации режима Fog computing).

Ключевые слова: схемотехническое моделирование; микроконтроллер; Proteus; нейронная сеть.

Поступила в редакцию 26.04.2022, принята к опубликованию 10.05.2022

**Investigation of a Neural Network Decomposition
by Proteus Design Suite**

V. V. Bakhtin¹, I. A. Podlesnykh², S. F. Tyurin³

¹Perm National Research Polytechnic University; Perm, Russia
e-mail: bakhtin_94@bk.ru; **ORCID:** 0000-0003-4873-7741, **AuthorID:** 882760

²Perm National Research Polytechnic University; Perm, Russia
e-mail: podlesnihwork@gmail.com; **ORCID:** 0000-0001-6260-393X, **AuthorID:** 1149281

³Perm National Research Polytechnic University; Perm, Russia
Perm State University; Perm, Russia
e-mail: tyurinsergfeo@yandex.ru; **ORCID:** 0000-0002-5707-5404, **AuthorID:** 629677

The division of a monolithic neural network into blocks with their implementation on programmable logic within the framework of the Fog computing concept is considered. It is assumed that considering possible re-configuration the implementation of blocks is performed on programmable logic: field-programmable gate array, FPGA (complex programmable logic device, CPLD), System-on-a-Chip, SoC or System-in-Package, SiP. The article explores such an implementation in the Proteus Design Suite based on ATmega32 microcontrollers. Modeling confirms the efficiency of the developed decomposition method. The research was carried out under the RFBR grant 20-37-90036 (Method of synthesizing neural network recognition devices for implementing the Fog computing mode)

Keywords: Circuit Simulation; Microcontroller; Proteus; Neural Network.

Received 26.04.2022, accepted 10.05.2022

DOI: 10.17072/1993-0550-2022-2-73-80



Эта работа © 2022 Бахтин В. В., Подлесных И. А., Тюрин С. Ф. лицензируется под CC BY 4.0. Чтобы просмотреть копию этой лицензии, посетите <http://creativecommons.org/licenses/by/4.0/>

Введение

Как известно, одна из важнейших проблем теории детерминированных конечных автоматов – проблема оптимального блочного синтеза, то есть разбиения исходного автомата на конечное число произвольно связанных между собой блоков (в общем виде до сих пор не решена [1]). В теории нейронных сетей рассматривается и решается более узкая задача разбиения монолитной нейронной сети на блоки [2–6], однако речь идет о последовательности блоков. Идея заключается в комплексировании этих блоков и "туманных" вычислений, которые, в отличие от "облачных" (cloud computing), образно говоря, приближены к "земле", к конкретному и часто неопытному пользователю, но остро нуждающемуся в нейросетевых технологиях. Противоречие в практике нейронных сетей, по мнению авторов, заключается в том, что разработанные в настоящее время мощные аппаратно-программные средства решения сложных практических задач не применимы вследствие отсутствия возможности их использования в "полевых" условиях. Это противоречие ярко высветила пандемия коронавируса и соответствующие трудности системы здравоохранения, медицинской диагностики, в том числе дистанционной, когда речь идет подчас о жизни и смерти пациента в условиях миллионов заражений в день. Другая сверхактуальная в настоящее время область – область применения в военной технике и аппаратуре, в экстремальных, боевых условиях с возможной реконфигурацией [7] после отказов, повреждений, что требует, конечно, отдельного рассмотрения. В то же время, несмотря на впечатляющие успехи нейроматематики, научно-методический аппарат декомпозиции нейронной сети, ориентированный на туманные вычисления, и позволяющий разрешить существующее практическое противоречие, в полной мере пока не разработан, что является, в свою очередь, противоречием в науке.

1. Метод и алгоритм декомпозиции

Для начала необходимо дать некоторые определения. Нейронную сеть, все слои которой производят свои вычисления на одном и том же вычислительном устройстве назовем *монолитной нейронной сетью (МНС)*. Если перед нами стоит задача получения нейронной сети, которая может проводить свои вы-

числения на нескольких связанных между собой устройствах, то нам потребуется разделить слои этой нейронной сети на блоки последовательных, идущих друг за другом слоев. Каждый из этих блоков будет выполнен на отдельном вычислительном устройстве, а промежуточные результаты будут переданы по сети между ними. Нейронную сеть, разбитую на набор подобных блоков, назовем *блочной нейронной сетью (БНС)*.

Сущность предложенного метода заключается в следующем:

1. Выбирается МНС, которую требуется декомпозировать; она переводится в предлагаемый формат;

2. Используется алгоритм декомпозиции МНС с заданными входными параметрами:

- количество устройств, на которые необходимо разделить нейронную сеть D ;
- массив вычислительных мощностей устройств $\{P_0, \dots, P_{D-1}\}$;
- флаг предпочитаемого способа разделения на блоки;

3. Получается D блоков БНС в файлах предлагаемого формата;

4. Переносится содержимое файла в программу для микроконтроллера;

5. Компилируется файл *.hex, он загружается как прошивка для устройства;

6. Запускается в работу каскад контроллеров, соединенных между собой.

Для реализации метода на процессорных устройствах с операционной системой вторая половина шагов видоизменится:

4. Переносится файл на устройство в соответствующую директорию файловой системы;

5. Запускается программа выполнения блока БНС, написанная на языке программирования высокого уровня;

6. Запускается в работу каскад устройств, соединенных между собой.

Возможен комбинированный вариант реализации каскада вычислительных устройств, содержащего как микроконтроллеры, так и процессорные устройства, в случае, когда между устройствами различного типа существует канал связи.

Существуют различные варианты входных параметров, в зависимости от которых способы разделения монолитной ИНС на блоки могут отличаться:

1. Разделение на блоки с равным количеством слоев на узле;
2. Разделение пропорционально производительности устройств (слои);
3. Разделение на блоки с равным количеством нейронов на узле;
4. Разделение пропорционально производительности устройств (нейроны);
5. Разделение с условием минимизации передаваемых по сети данных.

В данной статье описывается работа с уже обученными нейронными сетями, то есть к моменту разделения на блочные нейронные сети веса синапсов известны и зафиксированы. Это ограничение временное и преодолимо в будущем. Также рассматриваются именно многослойные нейронные сети, так как в полных нейронных сетях слишком много связей, и их разделение на независимые кластеры, которые могли бы выполняться на различных физических устройствах, является сложной задачей.

Кроме того, в данной статье рассматривается математическая модель разделения нейронной сети без обратных связей. В будущем в рамках исследования возможна доработка которая позволит разделять сети с обратными связями.

Процесс преобразования монолитной ИНС в каскад блочных ИНС, результат вычисления которого совпадает с результатами монолитной нейронной сети, назовем *декомпозицией* искусственной нейронной сети.

Количество слоев K , количество устройств D . Количество подсетей: D , по числу устройств.

Теорема о существовании блочной нейронной сети. Для любого $D \leq K$ существует последовательность нейронных сетей $\{\bar{X}_0, \dots, \bar{X}_{D-1}\}$, называемых блочными нейронными сетями, такая, что результат последовательного их вычисления, т.е. когда выходное значение $y_i^{(n)}$ нейронной сети \bar{X}_n является входным значением $x_i^{(n+1)}$ нейронной сети \bar{X}_{n+1} для $n \in [1, D - 1]$, совпадает с результатом работы исходной сети X для всех $x_i^{(0)} \in \mathbb{R}$.

База индукции. $D = 1$. В этом случае исходная нейронная и есть единственный элемент последовательности, т. е. вырождается сама в себя. Соответственно, результат работы такой последовательности из одного блока равен результату работы исходной нейронной сети.

Индукционный переход. Пусть утверждение справедливо для $D = n$. Тогда исходная сеть X декомпозируется на последовательность нейронных сетей $\{\bar{X}_0, \dots, \bar{X}_{n-1}\}$.

Докажем справедливость утверждения для $D = n+1$. Возьмем последовательность $\{\bar{X}_0, \dots, \bar{X}_{n-1}\}$ и разделим в ней нейронную сеть, в которой не менее двух слоев, на две нейронные сети. Таким образом получаем последовательность $\{\bar{X}_0, \dots, \bar{X}_{n+1-1}\}$, которая в конечном итоге будет записана в виде $\{\bar{X}_0, \dots, \bar{X}_n\}$. Если же среди нейронных сетей последовательности $\{\bar{X}_0, \dots, \bar{X}_{n-1}\}$ не нашлось ни одной нейронной сети с числом слоев $k > 1$, то это означает, что $n = K$. Из этого следует то, что достигнуто предельно допустимое состояние декомпозиции нейронной сети.

Теперь, когда доказано, что для каждой МНС существует последовательность нейронных сетей, являющихся ее декомпозицией, докажем, что результат работы исходной нейронной сети и каскада блочных нейронных сетей совпадет на одинаковом наборе входных данных.

Процесс преобразования каскада блочных ИНС в монолитную ИНС, результат вычисления которой совпадает с результатами блочной нейронной сети назовем *композицией* искусственной нейронной сети.

Теорема об эквивалентности монолитной и блочной НС. Результаты работы монолитной нейронной сети X и последовательности блочных нейронных сетей $\{\bar{X}_0, \dots, \bar{X}_{D-1}\}$, полученной декомпозицией монолитной нейронной сети X , равны для всех входных значений $x_i^{(0)} \in \mathbb{R}$.

Доказательство: в соответствии с алгоритмом декомпозиции, последний слой монолитной нейронной сети X с номером K и последний слой нейронной сети \bar{X}_{D-1} представляют собой один и тот же слой, что, очевидно, позволяет говорить об одинаковой размерности выходных векторов $\{y_0^K, \dots, y_{H_K-1}^K\}$ и $\{y_0^{L_{D-1}}, \dots, y_{H_K-1}^{L_{D-1}}\}$.

Аналогичным образом можно утверждать, что входные вектора для блочной и монолитной нейронной сети имеют одинаковую размерность H_0 .

Предположим противное: что на одинаковых исходных данных $\{x_0^0, \dots, x_{H_0-1}^0\}$ монолитная нейронная сеть X и последовательность нейронных сетей $\{\bar{X}_0, \dots, \bar{X}_{D-1}\}$ могут

дать различающиеся выходные векторы: $\{y_0^K, \dots, y_{H_{K-1}}^K\}$ и $\{y_0^{L_{D-1}}, \dots, y_{H_{K-1}}^{L_{D-1}}\}$.

Общая рекуррентная формула монолитной нейронной сети выглядит следующим образом:

$$X: \begin{cases} y_i^{(K)} = f_i^{(K)} \left(\sum_{j=0}^{H_{K-1}} w_{ij}^{(K)} \cdot y_i^{(K-1)} \right) \\ \dots \\ y_i^{(0)} = x_i^{(0)} \end{cases}$$

Количество слоев K , количество устройств D . Количество подсетей: D , по числу устройств. Для случая, когда $K \geq D$ и K кратно D , получим следующие рекуррентные формулы блочных нейронных сетей для каждого из устройств:

$$\begin{aligned} \overline{X_0}: & \begin{cases} y_i^{(\frac{K}{D})} = f_i^{(\frac{K}{D})} \left(\sum_{j=0}^{\frac{H_K}{D}-1} w_{ij}^{(\frac{K}{D})} \cdot y_i^{(\frac{K}{D}-1)} \right), \\ \dots \\ y_i^{(0)} = x_i^{(0)} \end{cases} \\ \overline{X_1}: & \begin{cases} y_i^{(\frac{2K}{D})} = f_i^{(\frac{2K}{D})} \left(\sum_{j=0}^{\frac{H_{2K}}{D}-1} w_{ij}^{(\frac{2K}{D})} \cdot y_i^{(\frac{2K}{D}-1)} \right), \\ \dots \\ y_i^{(\frac{K}{D}+1)} = y_i^{(\frac{K}{D})} \\ \dots \end{cases} \\ \overline{X_{D-1}}: & \begin{cases} y_i^{(\frac{DK}{D})} = f_i^{(\frac{DK}{D})} \left(\sum_{j=0}^{H_{K-1}} w_{ij}^{(\frac{DK}{D})} \cdot y_i^{(\frac{DK}{D}-1)} \right), \\ \dots \\ y_i^{(\frac{(D-1)K}{D}+1)} = y_i^{(\frac{(D-1)K}{D})}. \end{cases} \end{aligned}$$

Далее начнем осуществлять композицию блочных нейронных сетей: объединим $\overline{X_0}$ и $\overline{X_1}$, в результате композиции получим нейронную сеть:

$$\overline{X_{01}}: \begin{cases} y_i^{(\frac{2K}{D})} = f_i^{(\frac{2K}{D})} \left(\sum_{j=0}^{\frac{H_{2K}}{D}-1} w_{ij}^{(\frac{2K}{D})} \cdot y_i^{(\frac{2K}{D}-1)} \right) \\ \dots \\ y_i^{(0)} = x_i^{(0)} \end{cases}$$

Тогда композиция последовательности блочных нейронных сетей $\{\overline{X_0}, \dots, \overline{X_{D-1}}\}$ будет представлять собой следующую нейронную сеть:

$$\overline{X_{0(D-1)}}: \begin{cases} y_i^{(\frac{DK}{D})} = f_i^{(\frac{DK}{D})} \\ \left(\sum_{j=0}^{\frac{H_{DK}}{D}-1} w_{ij}^{(\frac{DK}{D})} \cdot y_i^{(\frac{DK}{D}-1)} \right) \\ \dots \\ y_i^{(0)} = x_i^{(0)} \end{cases}$$

После несложных преобразований получим следующий вид нейронной сети $\overline{X_{0(D-1)}}$:

$$\overline{X_{0(D-1)}}: \begin{cases} y_i^{(K)} = f_i^{(K)} \left(\sum_{j=0}^{H_{K-1}} w_{ij}^{(K)} \cdot y_i^{(K-1)} \right) \\ \dots \\ y_i^{(0)} = x_i^{(0)} \end{cases}$$

Таким образом очевидно, что исходная монолитная нейронная сеть X и результат композиции последовательности блочной нейронной сети $\overline{X_{0(D-1)}}$ представляют собой одну и ту же нейронную сеть.

На одном наборе входных данных представленные нейронные сети дадут одинаковые результаты вычислений, но по нашему предположению эти сети могут дать различные результаты вычислений. Возникает противоречие, а значит, верно обратное, что и требовалось доказать.

2. Создание моделей в Proteus Design Suite

Для экспериментов был выбран микроконтроллер ATmega32 бывшей компании Atmel, ныне Microchip Technology, так как, с одной стороны, у него имеется весь необходимый функционал: выполнение программ, написанных на языке C, наличие USART, встроенные Timer/Counter, возможности для подключения программаторов и подключения к переходникам UART-USB.

При этом он хорошо моделирует вычислительное устройство малой мощности, имеет невысокую частоту генератора тактов 8 МГц, а приобрести его физически не составляет сложностей.

Моделирование выполняется в пока еще свободно распространяемой системе схемотехнического моделирования (Design Suite) Proteus фирмы Labcenter Electronics (Великобритания) [8–11] (рис. 1):



Рис. 1. Значок Proteus Design Suite

Интересно, что в древнегреческой мифологии Протеус (Протей) – сын Посейдона и Геры, обладающий удивительными способностями перевоплощения, что как раз согласуется с используемыми в предложенной парадигме декомпозиции программируемыми логическими устройствами, допускающими реконфигурацию. Предполагается, что микроконтроллеры конфигурируются в нейросеть в специальных микросхемах систем на кристалле. Однако то же справедливо и для систем в пакете, а также для ПЛИС, то есть "программируемая логика" понимается обобщенно.

Создаем схему электрическую функциональную микроконтроллера ATmega32 в Design Suite Proteus фирмы Labcenter Electronics (рис. 2а). Подключим кварцевый резонатор X1 ко входам XTAL1, XTAL2 встроенного тактового генератора микроконтроллера, а также два конденсатора C1, C2 для его "раскачки" (рис. 2б).

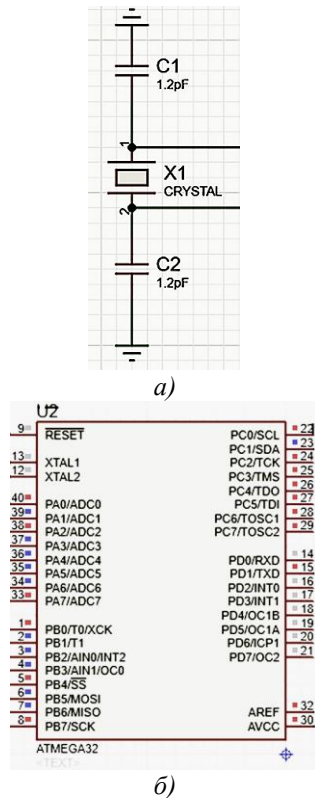


Рис. 2. Микроконтроллер: а) модель кварцевого резонатора для встроенного тактового генератора ATmega32; б) модель микросхемы ATmega32

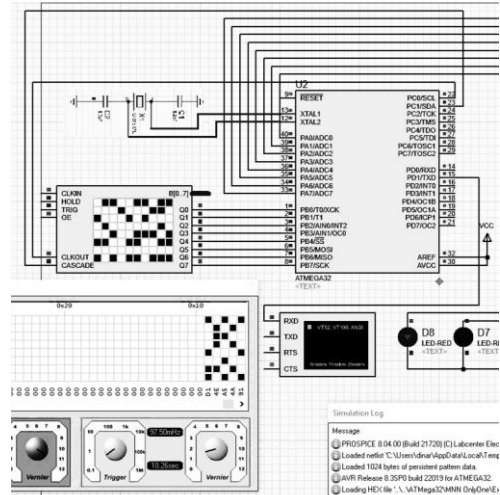


Рис. 3. Схема электрическая функциональная микроконтроллера ATmega32 в Design Suite Proteus фирмы Labcenter Electronics

Используем для задания тестовой входной байтовой информации нейросети генератор слов PATTERN GENERATOR, подключим его к порту В (рис. 3). Генератор слов может рассматриваться и как преобразователь сигналов из аналоговой формы в цифровую. Так, каждая пара битов моделирует четыре уровня аналогового сигнала. Байт моделирует 256 уровней. Возможен анализ многобайтного входа, а также использование встроенного АЦП.

К порту А подключим светодиоды D1-D8 для бинарной индикации результатов работы нейросети. Виртуальный терминал подключаем к PD1/TXD (трансмиссивтер UART).

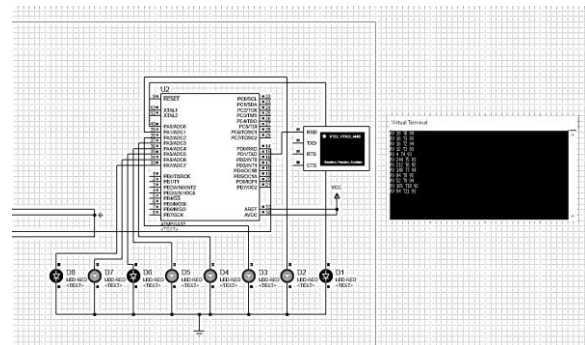


Рис. 4. Последний микроконтроллер в схеме на 3 микроконтроллера

Далее схемы подключения кварцевых резонаторов ко входам встроенного тактового генератора микроконтроллера с целью упрощения рисунка не будем указывать.

Запускаем моделирование, оно выполняется успешно. Создаем вторую схему в Proteus с декомпозицией нейросети по трем микроконтроллерам (рис. 4).

Результаты работы первого микроконтроллера будут поступать на второй, а второго – на третий микроконтроллер по стандартному последовательному интерфейсу. Моделирование схемы по рис. 4 также проведено успешно.

Создаем и моделируем схему в Proteus с декомпозицией нейросети по пяти микроконтроллерам (рис. 5).

Предлагаемым методом синтеза было проведено моделирование БНС пятью способами, в каждом из которых было проведено разбиение на три размерности каскадов: три, пять и семь микроконтроллеров, а также запуск эталонной схемы с одним вычислительным устройством.

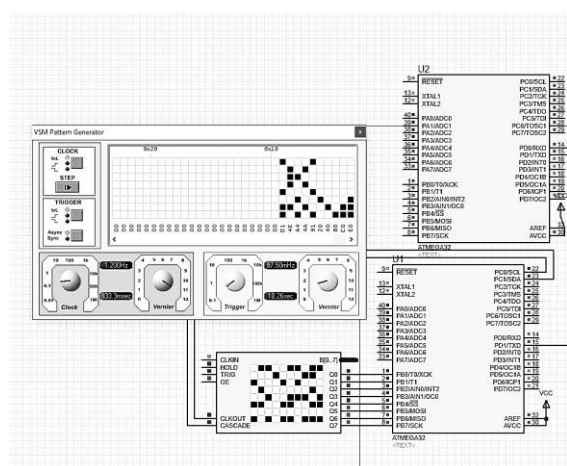


Рис. 5. Первые два микроконтроллера в схеме на 7 микроконтроллерах

В каждом из этих экспериментов на вход подавались 12 различных векторов входных параметров для каждой из собранных электрических функциональных схем.



Рис. 6. Результаты выполнения алгоритма БНС и замеры числа тиков таймера, потребовавшихся для выполнения вычислений

Аналогично могут быть построены схемы с большим числом микроконтроллеров. В микроконтроллеры на этапе программирования заносятся полученные с помощью пред-

ложенного алгоритма блоки нейросети. Моделирование на различных тестовых массивах входных бинарных наборов схем выполнено успешно.

3. Оценка результатов моделирования

При частоте 8 МГц один "тик" таймера микроконтроллера (Tick clock – понятие, связанное с обработкой прерываний) будет составлять порядка 92 мкс.

Первой задачей данного исследования является эмпирическое, при помощи инструментов схемотехнического моделирования, подтверждение теоремы об эквивалентности монолитной и блочной нейронных сетей.

Полученные результаты моделирования показывают, что представленная теорема об эквивалентности монолитной и блочной НС, доказанная выше, подтверждается также эмпирическими данными. Результирующие вектора, полученные на одинаковых данных монолитной нейронной сетью и каждой из блочных нейронных сетей, созданных из исходной с помощью декомпозиции, совпадают между собой. В процессе работы удалось также эмпирически подтвердить теорему о существовании блочной нейронной сети, создав из исходной монолитной нейронной сети блочные нейронные сети различных размерностей.

В рамках моделирования и проведения экспериментов по запуску блочных нейронных сетей осуществлялась декомпозиция пяти различных монолитных нейронных сетей. Каждая из них была декомпозирована на каскады из трех, пяти и семи вычислительных устройств. На вход блочной нейронной сети в каждом из моделирований направлялись пятьдесят одинаковых векторов входных параметров. Ниже представлены результаты моделирования нейронной сети, изначально состоявшей из тридцати слоев нейронов (рис. 7).

Выходной набор нейросети (один микроконтроллера)	Декомпозиция на три микроконтроллера	Декомпозиция на пять микроконтроллеров	Декомпозиция на семь микроконтроллеров	Counter ticks
0x13	0x13	0x13	0x13	
0x12	0x12	0x12	0x12	
0x10	0x10	0x10	0x10	
0x0c	0x0c	0x0c	0x0c	
0x04	0x04	0x04	0x04	
0xf4	0xf4	0xf4	0xf4	
0xd4	0xd4	0xd4	0xd4	
0x94	0x94	0x94	0x94	
0x54	0x54	0x54	0x54	
0x34	0x34	0x34	0x34	
0x24	0x24	0x24	0x24	
210	155	90	50	

Рис. 7. Результаты тестирования декомпозиции нейронной сети

Количество проведенных моделирований и испытаний блочных нейронных сетей позволяет считать данный метод эффективно решающим поставленные задачи. Среди 20 моделирований блочных нейронных сетей 20 дали результат, совпадающий с исходной монолитной нейронной сетью, что позволяет говорить об эффективности в 100 %. Кроме того, результаты моделирования показывают, что каскад из блочных нейронных сетей сохраняет основные параметры монолитной нейронной сети, из которой он был декомпозирован. Например, Точность (Acc, Accuracy) – это способность модели давать правильный результат относительно общего количества исследований. Это происходит потому, что блочная нейронная сеть сохраняет все слои нейронов и связи между ними, которые были в исходной сети.

Второй задачей данного исследования является подтверждение утверждения о том, что разделение монолитной нейронной сети на блочную, с последующим вычислением результатов работы блочной нейронной сети на каскаде устройств, позволяет снизить нагрузку на единичное вычислительное устройство в данном каскаде.

Данное утверждение подтвердилось в результате проведения схемотехнического моделирования. Например, в случае каскада из 7 устройств, вычисления на единичном устройстве в каскаде занимают 50 тиков, тогда как в случае вычисления результатов монолитной нейронной сети на единичном устройстве занимают 210 тиков. Следовательно, нагрузка на устройство сокращается в 4 раза.

Выводы

Таким образом, решена задача разработки научно-методического аппарата синтеза устройств реализации искусственных нейронных сетей, ориентированных на туманные вычисления и использующих программируемую логику. Создана математическая модель блочной нейронной сети, сформулированы и доказаны необходимые теоремы, разработаны алгоритмы декомпозиции и работы блочных нейронных сетей. Полностью сформулирован метод синтеза устройств нейросетевого распознавания, ориентированных на туманные вычисления, проведено схемотехническое моделирование работы блочных нейронных сетей в САПР Proteus.

Моделирование подтвердило достоверность результатов исследования.

Предполагается использование для декомпозиции не только персональных компьютеров, смарт часов, смартфонов и других гаджетов, но и более простых устройств, содержащих недорогие аппаратные средства программируемой логики.

Например, медицинские приборы типа термометров, измерителей давления и пр., поскольку, например, дорогой "интернет вещей" может быть не у всех пациентов, да и не у всех медицинских работников, особенно "в глубинке". В то же время предлагаемый подход позволяет в перспективе перевести на новый уровень и некоторые системы, используемые в вооружении и военной технике, в которых ранее реализация нейронных сетей была невозможна.

Список литературы

1. Кузнецов О.П. Дискретная математика для инженера. Сер. Учебники для вузов. Специальная литература (3-е изд., перераб. и доп.). СПб. [и др.], 2009.
2. Руднев В.А. Применение микроконтроллеров для реализации нейронных сетей // Вестник Южно-Уральского государственного университета. Серия: компьютерные технологии, управление, радиоэлектроника, 2012. № 23. С. 181–183.
3. Novac P.E. et al. Quantization and deployment of deep neural networks on microcontrollers. *Sensors*, 2021. Т. 21, № 9. С. 2984. DOI: 10.3390/s21092984.
4. Cotton N.J., Wilamowski B.M., Dundar G. A neural network implementation on an inexpensive eight-bit microcontroller. 2008 International Conference on Intelligent Engineering Systems, 2008. С. 109–114. DOI: 10.1109/INES.2008.4481278.
5. Tu Y. et al. A power efficient neural network implementation on heterogeneous FPGA and GPU devices. 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), 2019. С. 193–199. DOI: 10.1109/IRI.2019.00040.
6. Goudarzi M. et al. An application placement technique for concurrent IoT applications in edge and fog computing environments. *IEEE Transactions on Mobile Computing*, 2020. Т. 20, № 4. С. 1298–1311, DOI: 10.1109/TMC.2020.2967041.

7. Угрюмов Е.П. Цифровая схемотехника: учеб. пособие для вузов. Санкт-Петербург: БХВ-Петербург, 2010. 797 с.
8. Proteus Downloads. URL: <https://www.labcenter.com/downloads/> (дата обращения: 23.03.2022).
9. Тюрин С.Ф., Ковыляев Д.А., Данилова Е.Ю., Городилов А.Ю. Программирование микроконтроллеров с использованием IDE: учеб. пособие / под ред. С.Ф. Тюрин. Пермь: Изд-во Перм. нац. исслед. политех. ун-та, 2021. 100 с.
10. Тюрин С.Ф. Вычислительная техника и информационные технологии / Рук-во к лаборатор. работам в системе Proteus 7.2. Пермь: Из-во Перм. гос. техн. ун-та, 2010. 135 с.
11. Тюрин С.Ф., Ковыляев Д.А., Данилова Е.Ю., Городилов А.Ю. Изучение программирования микроконтроллеров в САПР PROTEUS // Вестник Пермского университета. Математика. Механика. Информатика. 2021. Вып. 2(53). С. 69–74.
5. Tu Y. et al. A power efficient neural network implementation on heterogeneous FPGA and GPU de-vices. 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), 2019. P. 193–199. DOI:10.1109/IRI.2019.00040.
6. Goudarzi M. et al. An application placement technique for concurrent IoT applications in edge and fog computing environments. IEEE Transactions on Mobile Computing, 2020. Vol. 20, No 4. P. 1298–1311. DOI: 10.1109/TMC.2020.2967041.
7. Ugryumov E.P. Digital Circuitry: a textbook for universities. St. Petersburg: BHV-Petersburg, 2010. 797 p.
8. Proteus Downloads [Electronic resource]. URL: <https://www.labcenter.com/downloads/> (accessed: 03.23.2022).
9. Tyurin C.F., Kovylyayev D.A., Danilova E.Y., Gorodilov A.Y. Programming of microcontrollers using IDE: textbook. manual / edited by S.F. Tyurin. Perm: Publishing house of Perm national research polytechnic university, 2021. 100 p.
10. Tyurin S.F. Computer engineering and information technologies. Guide to laboratory work in the Proteus 7.2 system. Perm: publishing house of Perm state technical university, 2010. 135 p.
11. Tyurin S.F., Kovylyayev D.A., Danilova E.Y., Gorodilov A.Y. Study of microcontroller programming in PROTEUS CAD // Bulletin of Perm University. Mathematics. Mechanics. Computer science. 2021. Issue 2(53). P. 69–74.

References

1. Kuznetsov O.P. Discrete mathematics for an engineer. Ser. Textbooks for universities. Special literature (3rd ed., reprinted and enhanced). SPb. [et al.], 2009.
2. Rudnev V.A. The use of microcontrollers for the implementation of neural networks. Bulletin of the South Ural State University. Russia: Computer Technologies, Management, Radioelectronics, 2012. No. 23. P. 181–183.
3. Novac P. E. et al. Quantization and deployment of deep neural networks on microcontrollers. Sensors, 2021. Vol. 21, No. 9. P. 2984. DOI: 10.3390/s21092984.
4. Cotton N.J., Wilamowski B.M., Dundar G. A neural network implementation on an inexpensive eight-bit microcontroller. 2008 International Conference on Intelligent Engineering Systems, 2008. P. 109–114. DOI:10.1109/INES.2008.4481278.

Просьба ссылаться на эту статью:

Бахтин В.В., Подлесных И.А., Тюрин С.Ф. Исследование декомпозиции нейронной сети в системе схемотехнического моделирования Proteus // Вестник Пермского университета. Математика. Механика. Информатика. 2022. Вып. 2(57). С. 73–80. DOI: 10.17072/1993-0550-2022-2-73-80.

Please cite this article as:

Bakhtin V.V., Podlesnykh I.A., Tyurin S.F. Investigation of a Neural Network Decomposition by Proteus Design Suite // Bulletin of Perm University. Mathematics. Mechanics. Computer Science. 2022. Issue 2(57). P. 73–80. DOI: 10.17072/1993-0550-2022-2-73-80.