



Международный журнал информационных технологий и энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004

ИСПОЛЬЗОВАНИЕ БИБЛИОТЕКИ BRAINJS ДЛЯ РАЗРАБОТКИ НЕЙРОННОЙ СЕТИ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVASCRIPT

Кочитов М.Е., Баженов Р.И.

ФГБОУ ВО Приамурский государственный университет имени Шолом-Алейхема, Биробиджан, Россия (679015, г. Биробиджан, ул. Широкая, д.70а), e-mail: r-i-bazhenov@yandex.ru.

В данной статье рассматривается разработка искусственной нейронной сети с помощью библиотеки BrainJS, написанной на языке программирования JavaScript. В этой библиотеке имеются различные методы и параметры по обучению нейронных сетей для определенных задач: распознавание изображений и фотографий, улучшение искусственного интеллекта, определение оптимального маршрута и другие. Также в статье была разработана на языке программирования JavaScript, на языке разметки HTML и на CSS-фреймворке Bootstrap веб-ориентированная программа Веб нейронная сеть (ВНС), позволяющая создавать и обучать искусственные нейронные сети на готовом наборе данных, используя для такой цели готовую JavaScript библиотеку BrainJS. Результат работы программы Веб нейронная сеть (ВНС) показал хороший результат обученной нейронной сети с программой Neural Network Wizard, выполняющей такую же работу.

Ключевые слова: искусственная нейронная сеть, библиотека, BrainJS, JavaScript, CSS-фреймворк, Bootstrap, HTML, обучение нейронной сети, готовый набор данных, Веб нейронная сеть, ВНС, Neural Network Wizard.

USING THE BRAIN LIBRARY FOR THE DEVELOPMENT OF A NEURAL NETWORK IN THE JAVASCRIPT PROGRAMMING LANGUAGE

Kochitov M.E., Bazhenov R.I.

FSBEI HE Sholom-Aleichem Priamursky State University, Birobidzhan, Russia (679015, Birobidzhan, street Shirokaya, 70a), e-mail: r-i-bazhenov@yandex.ru

This paper discusses the development of an artificial neural network using the BrainJS library, written in the JavaScript programming language. This library has various methods and parameters for training neural networks for specific tasks: recognition of images and photographs, improving artificial intelligence, determining the optimal route, and others. The article also developed a web-oriented program Web neural network (ANS) in the JavaScript programming language, in the HTML markup language and in the Bootstrap CSS framework, which allows you to create and train artificial neural networks on a ready-made data set using a ready-made JavaScript library for this purpose BrainJS. The result of the Web Neural Network (ANS) program showed a good result of a trained neural network with the Neural Network Wizard program that does the same job.

Keywords: artificial neural network, library, BrainJS, JavaScript, CSS framework, Bootstrap, HTML, neural network training, ready data set, Web neural network, ANS, Neural Network Wizard.

В настоящее время искусственные нейронные сети пользуются большой популярностью в различных веб-технологиях. Они выполняют задачи, подобные тому, какие выполняет сам человеческий мозг, то есть они предназначены для того, чтобы их можно было обучать различными данными и в дальнейшем проверять результаты обученной нейронной сети. Иными словами проверка результатов происходит так, чтобы убедиться насколько искусственная нейронная сеть хорошо запомнила данные, которыми она обучалась. Нейронные сети в большинстве случаев используются в основном на обучении набором данных, в распознавании изображений и фотографий, в построении оптимального маршрута, в робототехнике и других.

В статье Н. Lin и С. Wang рассматривается влияние распределения электромагнитного излучения на нелинейную динамику нейронной сети с n нейронами. Также в статье была разработана и проанализирована новая математическая модель нейронной сети под воздействием электромагнитного излучения, в которой электромагнитное излучение эквивалентно магнитному потоку, проходящему через клеточную мембрану. Хаотическую динамику нервной системы детально изучают, стимулируя разное количество нейронов в модели нейронной сети, состоящей из трех нейронов [1].

Рассматривая статью Х. Wang, М. Che и Y. Wei можно увидеть разработку двух моделей нейронной сети для решения неособой многолинейной тензорной системы. Теоретический анализ показывает, что каждая из моделей нейронных сетей обеспечивает производительность сходимости. Для возможной аппаратной реализации предлагаемых моделей нейронных сетей, основанных на цифровых схемах, мы принимаем правило разности типа Эйлера для дискретизации соответствующих моделей градиентных нейронных сетей (GNN) [2].

Авторы А. Taherkhani, А. Belatreche, Y. Li, G. Cosma, L. P. Maguiree и Т.М. McGinnitaye в своей статье рассмотрели искусственные нейронные сети, которые используются в качестве мощного инструмента обработки в различных областях, таких как распознавание образов, управление, робототехника и биоинформатика. Их широкое применение побудило исследователей улучшать искусственные нейронные сети путем исследования биологического мозга. Неврологические исследования значительно продвинулись в последние годы и продолжают раскрывать новые характеристики биологических нейронов [3].

В статье Н. Sahiner и Х. Liu рассматривается искусственная нейронная сеть, являющаяся привлекательным методом для анализа сложных гамма-спектров, полученных при активации нейронов. Это исследование предлагает улучшенную методологию для анализа гамма-спектров активации нейронов с использованием искусственной нейронной сети. Искусственная нейронная сеть для анализа смоделированных гамма-спектров привела к расчетным концентрациям элементов [4].

Рассматривая статью S. Neo и J. H. Lee можно заметить параллельную архитектуру нейронной сети для повышения производительности анализа нелинейных главных компонентов на основе нейронных сетей. Существуют два типичных подхода для такого анализа: одновременное выделение основных компонентов с использованием одной автоассоциативной нейронной сети (также известной как автоэнкодер) и последовательное выделение с использованием нескольких нейронных сетей последовательно. Предлагаемая архитектура может быть получена путем систематического сокращения сетевых подключений

полностью связанной автоассоциативной нейронной сети, что приводит к разделению нейронных сетей [5].

D. Ayala, A. Borrego, I. Hernández и D. Ruiz в своей статье рассмотрели три вклада: во-первых, понимание архитектурных проблем, возникающих при использовании нейронных сетей для семантической маркировки; во-вторых, новая реализация семантической маркировки, которая использует современный классификатор нейронных сетей, который достигает значительно лучших результатов, чем другие четыре традиционных классификатора; в-третьих, сравнение результатов, полученных прежней сетью, при использовании различных поднаборов признаков, сравнение текстовых признаков со структурными и доменно-зависимых с независимыми от домена [6].

В статье T. Li, Z. Zhang и H. Chen рассматривается архитектура нейронной сети, которая может использовать последовательность последовательностей изображений пламени с динамическим пространственно-временным соотношением для прогнозирования состояния сгорания вращающейся печи. Предложенная архитектура нейронной сети реализует сквозную модель вывода, которая непосредственно опирается на входные данные, тем самым устраняя необходимость в традиционных сложных процедурах извлечения признаков. Этот метод объединяет преимущества сверточных нейронных сетей (CNN) и рекуррентных нейронных сетей (RNN) для облегчения эффективного прогнозирования [7].

Рассматривая статью C. Jiang и Q. Chen можно увидеть традиционную слепую модель восстановления изображения с супер-разрешением на основе нейронной сети, которая легко попадает в проблему локального минимума, поэтому создается слепая модель восстановления изображения с супер-разрешением на основе хаотической нейронной сети. Во-первых, шум искаженного изображения удаляется с помощью вейвлет-преобразования, чтобы уменьшить влияние шума ухудшенного изображения на результат слепого восстановления. Затем создается упрощенная модель хаотической нейронной сети путем введения переходного хаоса и изменяющегося во времени приращения в хаотическую нейронную сеть [8].

Авторы W. Li, M. Li, J. Qiao и X. Guo в своей статье рассмотрели повышение производительности моделирования нелинейных систем, которая предлагает адаптивную модульную нейронную сеть (FC-AMNN) на основе кластеризации, моделирующая механизм обработки информации человеческого мозга таким образом, что разные данные обрабатываются разными модулями параллельно. Во-первых, объекты кластеризуются с использованием алгоритма адаптивной кластеризации признаков, и количество модулей в FC-AMNN определяется количеством кластеров объектов автоматически [9].

В статье A. Patiño-Saucedo, H. Rostro-Gonzalez, T. Serrano-Gotarredona и B. Linares-Barranco рассматриваются нейронные сети, которые позволили добиться значительных успехов в последнее время благодаря главным образом улучшенным возможностям параллельных вычислений в соответствии с законом Мура, что позволило сократить время, необходимое для изучения параметров сложных многоуровневых нейронных архитектур. В этой статье рассматривается, что на нейроморфном оборудовании возможна реализация современных моделей как для MNIST, так и для наборов данных распознавания цифр NMNIST на основе событий. Также используется два подхода: путем непосредственного преобразования классической нейронной сети в ее заостренную версию и путем обучения зарождающейся сети с нуля [10].

Рассматривая статью А. Choi, R. Wang и А. Darwiche можно заметить идентификацию некоторых ключевых различий между функциями, вычисляемыми нейронными сетями, и функциями по маргинальным запросам байесовской сети, показывая, что первые более выразительны, чем вторые. Кроме того, было предложено простое дополнение к байесовским сетям (оператор тестирования), которое позволяет их маргинальным запросам стать «универсальными аппроксиматорами» [11].

Y. Dong, Z. Fu, Y. Peng, Y. Zheng и X. Li в своей статье рассмотрели метод точного оплодотворения кукурузы на основе нейронной сети вейвлет-BP. Во-первых, существующие экспериментальные данные «3414» были выбраны и интерполированы в качестве рабочих данных для моделирования. Затем общее состояние низкой частоты и детали высокой частоты выхода были рассчитаны с помощью метода вейвлет-разложения и реконструкции. После этого была создана нейронная сеть вейвлет-BP [12].

В статье М. Kobayashi рассматривается комплексная нейронная сеть Хопфилда (CHNN) представляющая собой многоуровневую модель нейронной сети Хопфилда и применяемая для хранения многоуровневых данных. Гиперболическая нейронная сеть Хопфилда (HNHN) является еще одной многоуровневой моделью. Емкость хранилища является важной проблемой нейронных сетей Хопфилда [13].

Рассматривая статью С. J. Z. Aguilar, J. F. Gómez-Aguilar, V. M. Alvarado-Martínez и Н. М. Romero-Ugalde можно увидеть объединение обоих подходов, чтобы предложить нейронную сеть дробного порядка (FONN) для идентификации системы. Алгоритм обучения был обобщен с учетом дробной производной Грюнвальда-Летникова. Этот новый подход к моделированию черного ящика подтверждается идентификацией трех различных систем (двух эталонных систем и реальной системы). Сравнение с другими подходами показало, что предложенная модель FONN достигла большей точности с меньшим количеством параметров [14].

Авторы G. Sideratos, А. Ikononopoulos и N. D. Hatziargyriou в своей статье рассмотрели предлагаемую реализацию, которая объединяет атрибуты ансамблевого прогнозирования, искусственных нейронных сетей и архитектур глубокого обучения. Предложенная модель первоначально группирует входные данные, используя новый метод нечеткой кластеризации для создания ансамблевого прогнозирования. Для каждого созданного кластера применяется новый регрессионный подход для локального моделирования задачи прогнозирования нагрузки [15].

Целью данного исследования является разработка веб-программы, которая способна создавать и обучать нейронные сети с использованием JavaScript библиотеки BrainJS.

BrainJS [16] – это готовая JavaScript библиотека в которой можно создавать и обучать искусственные нейронные сети разными методами. Самый популярный метод это обратное распространение ошибки (MSE), благодаря которому можно обучать нейронную сеть много раз, пока ее среднеквадратичная ошибка не станет очень маленькой. Данная библиотека поддерживает создание и обучение обычных и рекуррентных нейронных сетей. Также при создании нейронной сети можно дополнительно задать параметры обучения: количество эпох, порог ошибки, функция активации, количество скрытых слоев и нейронов в каждом слое и т.д. Сама библиотека может использоваться в различных областях к применению нейронных сетей на текстах, наборах данных, в распознавании изображений, фотографий, в оптимизации маршрутов и т.д.

Веб-ориентированная программа по созданию и обучению искусственных нейронных сетей будет разрабатываться на следующих инструментах:

- готовая библиотека нейронных сетей BrainJS [16],
- язык программирования JavaScript,
- язык разметки HTML,
- CSS-фреймворк Bootstrap [17].

Разрабатываемая веб-программа будет иметь интерфейс пошаговой настройки и обучения нейронной сети с прикреплением обучающей выборки данных в виде таблицы. Сама веб-программа для начала будет предлагать пользователю добавить в текстовое поле таблицу набора данных. Далее указать нормализацию данных и тип данных – входной и целевой. После необходимо указать настройки обучения нейронной сети и напоследок запустить процесс обучения нейронной сети и проверять результаты сети на основе данных, на которых она обучилась.

Рассмотрим разработанную программу «Веб нейронная сеть» (ВНС) [21], которая предназначена создавать и обучать искусственные нейронные сети на обучающей выборке данных.

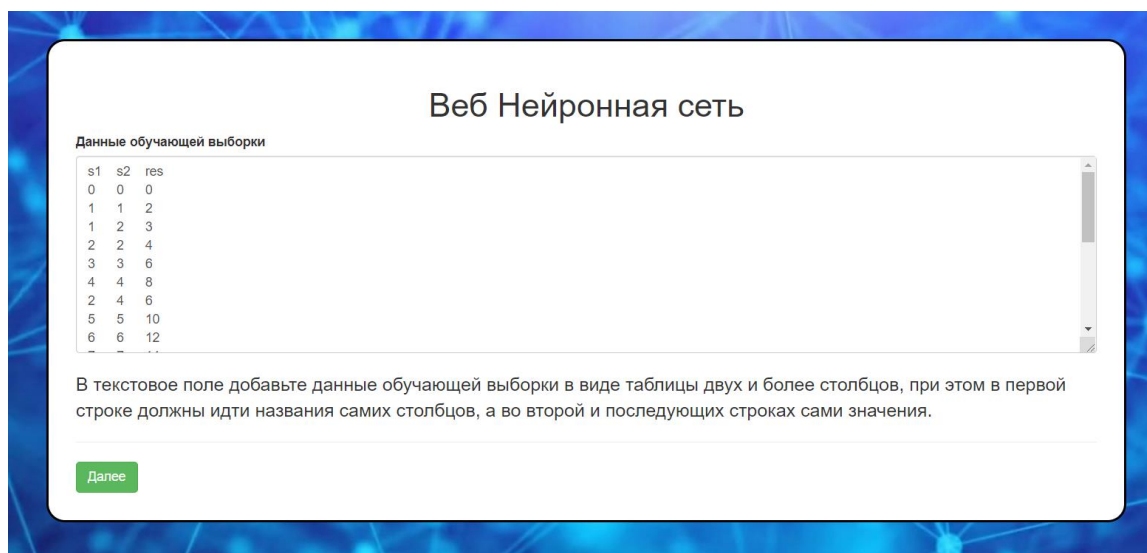


Рисунок 1 - Данные обучающей выборки

На рисунке 1 изображен первый шаг веб-программы «ВНС», это добавление данных обучающей выборки в виде таблицы. В первой строке обозначаются названия данных каждого столбца, в последующих строка идут сами значения данных. На примере были взяты данные сложения простых чисел, то есть $s1$ – первое слагаемое, $s2$ – второе слагаемое, res – сумма двух слагаемых. Веб-программа «ВНС» поддерживает неограниченное число данных, как самих строк, так и самих столбцов.

Веб Нейронная сеть

Список доступных полей

- s1
- s2
- ges

Использовать поле как:

- Входное
- Целевое
- Не использовать

Нормализовать поле как:

- (X-MIN) / (MAX-MIN)

Параметры нормализации:

MIN = 0
MAX = 10

Назад Далее

Рисунок 2 - Список доступных полей

На рисунке 2 показан второй шаг веб-программы «ВНС» это список доступных полей (первая строка обучающей выборке данных на прошлом шаге) и опции для использования их как входное, целевое или не используемое. Опция нормализации обязательно должна присутствовать, так как нейронная сеть работает со значениями в диапазоне от 0 до 1 и если в таблице значения выходят за данный диапазон, то необходимо их нормализовать по формуле: $(X-MIN) / (MAX-MIN)$, где X – текущее значение, MIN – минимальное значение из доступного поля (столбца), MAX – максимальное значение из доступного поля (столбца). В параметрах нормализации на интерфейсе веб-программы «ВНС» автоматически определяется, какое минимальное и максимальное значение.

Веб Нейронная сеть

Параметры нейронной сети:

Число нейронов во входном слое: 2
Число нейронов в выходном слое: 1
Число скрытых слоев нейронной сети: 1

Число нейронов в скрытом слое 1: 10

Назад Далее

Рисунок 3 - Параметры нейронной сети

На рисунке 3 отображен третий шаг веб-программы «ВНС» – параметры нейронной сети, где указывается количество скрытых слоев и количество нейронов в каждом скрытом слое.

Чем больше скрытых слоев и нейронов, тем лучше нейронная сеть будет обучаться, и запоминать большое количество данных.

Веб Нейронная сеть

Параметры обучения:

Скорость обучения	Момент
<input type="text" value="0,3"/>	<input type="text" value="0,9"/>
Количество эпох (итераций)	Порог ошибки
<input type="text" value="10000"/>	<input type="text" value="0,0001"/>

Рисунок 4 - Параметры обучения нейронной сети

На рисунке 4 продемонстрирован четвертый шаг веб-программы «ВНС», в котором находятся четыре важных параметра обучения: скорость обучения, момент, количество эпох (итераций) и порог ошибки. Длительность по времени обучения зависит только от количества скрытых слоев и от количества эпох (итераций), то есть чем больше указано значение, тем дольше будет обучаться нейронная сеть, и займет много времени.

Веб Нейронная сеть

Конфигурация нейронной системы:

Входные поля:
s1
s2

Целевые поля:
ges

Число входных нейронов: 2
Число выходных нейронов: 1
Число скрытых слоев нейронной сети: 1
Слой 1 - число нейронов: 10

Рисунок 5 - Конфигурация нейронной системы.

На рисунке 5 показан пятый шаг веб-программы «ВНС», показывающий конфигурацию нейронной системы, которая была настроена в предыдущих шагах. Окно в этом шаге представляет из себя отчет настроек нейронной сети перед ее обучением. Если что-то неверно

указано в конфигурации, то можно вернуться на предыдущие шаги и поменять некоторые параметры.

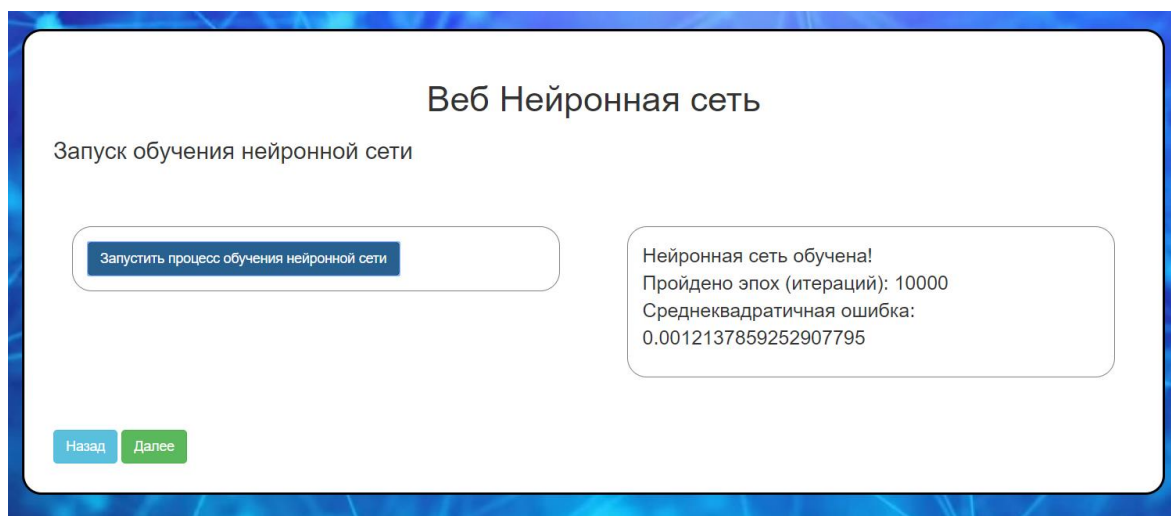


Рисунок 6 - Процесс обучения нейронной сети

На рисунке 6 изображен шестой шаг веб-программы «ВНС», который дает пользователю при нажатии синей кнопки запустить процесс обучения и нейронной сети и после обучения выдать в правом окошке информацию о количестве пройденных эпох и вычисленной среднеквадратичной ошибки.

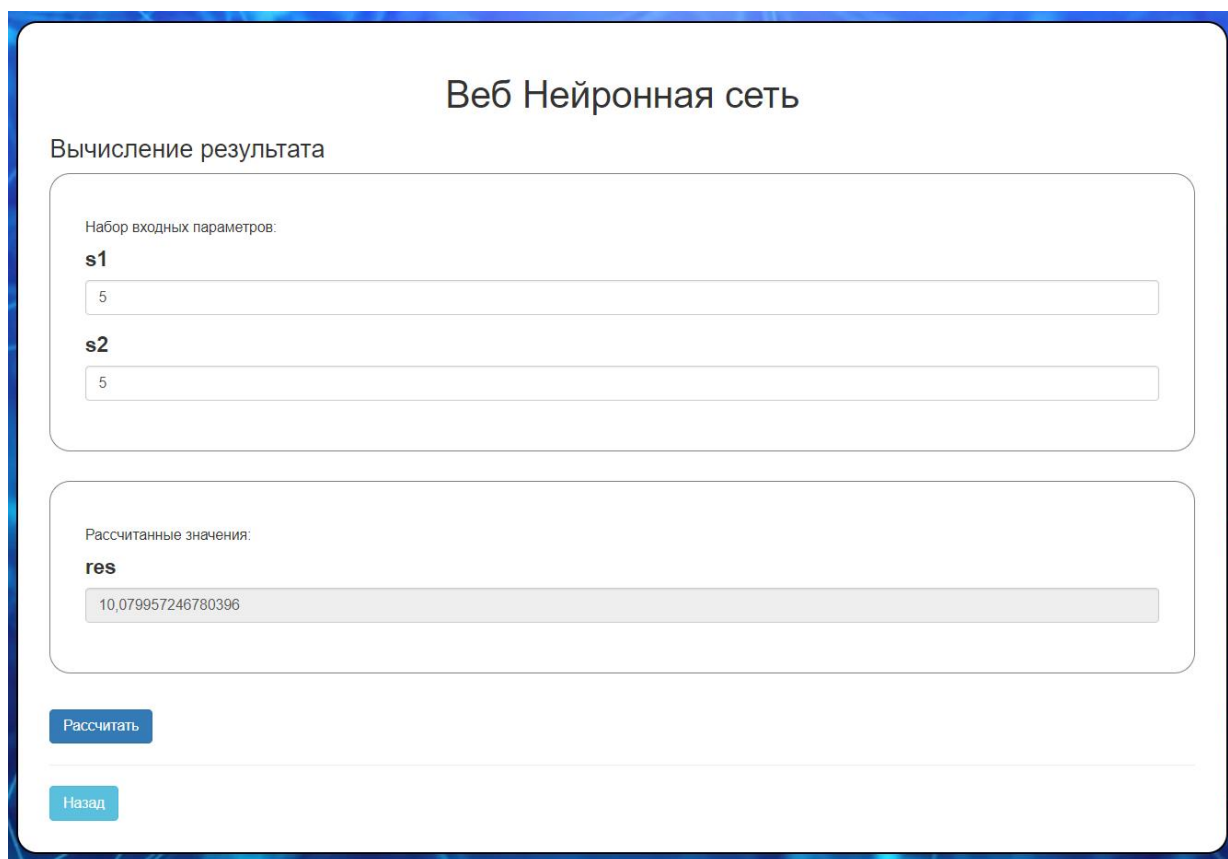


Рисунок 7 - Вычисление результата нейронной сети

На рисунке 7 представлен последний седьмой шаг веб-программы, который дает пользователю вычислять результат работы нейронных сетей, вводя входные параметры и получая самой нейронной сетью рассчитанные значения. Веб-программа при входных параметрах: $s1 = 5$, $s2 = 5$ выдает результат приблизительно 10.079, это означает, что нейронная сеть хорошо обучилась и выдает правильный результат сложения двух входных параметров: $5 + 5 = 10$.

Следующим этапом было предложено проверить результаты веб-программы «ВНС» на других входных параметрах (таблица 1)

Таблица 1 – Результаты обученной нейронной сети

Входные данные		Выходные данные	
<i>s1</i>	<i>s2</i>	<i>res</i>	<i>погрешность</i>
10	8	17.9	-0.1
5	7	12.4	0.4
3	5	7.6	-0.4
2	2	3.7	-0.3
7	2	9.4	0.4

В таблице 1 показаны результаты нейронной сети разных входных данных ($s1$ – первое слагаемое, $s2$ – второе слагаемое, то есть $s1 + s2$). Из таблицы результатов некоторые выходные данные (res – сумма двух слагаемых, то есть $s1 + s2 = res$) имеют погрешности с правильными данными, это связано с тем, что нейронная сеть на некоторых входных данных не так хорошо обучилась. Ей необходимо для переобучения еще больше количества эпох и больше количества скрытых слоев, как и самих нейронов, чтобы нейронная сеть имела очень маленькую ошибку на всех данных обучающей выборки.

Разработанная веб-программа «ВНС» [21] была составлена подобным образом аналогичной программе под названием «Neural Network Wizard» [22], именно под нее был составлен пошаговый упрощенный интерфейс с аналогичной работой обучения нейронной сети. Далее загрузим в веб-программу «ВНС» и в программу «Neural Network Wizard» данные квадратного корня, после этого настроим идентично параметры и проверим результат работы обеих программ.

На рисунках 8 и 9 изображены две разные программы: веб-программа «ВНС» и программа «Neural Network Wizard». Результат вычисления: это квадратный корень из 4, который равен 2. Веб программа выдала результат – 2.15, а программа «Neural Network Wizard» выдала результат – 1.9, то есть погрешность от правильного ответа равна 0.1 в меньшую и большую сторону.

Теперь рассмотрим вычисления квадратного корня на 16 входных данных и проверим результаты работы двух программ (таблица 2)

В таблице 2 представлены результаты вычислений квадратного корня у двух программ: веб-программы «ВНС» и «Neuron Network Wizard». На всех 16-ти входных данных обе программы рассчитывали результат по-разному, опираясь на разные погрешности. Однако в некоторых входных данных обе программы вычислили правильный результат квадратного корня.

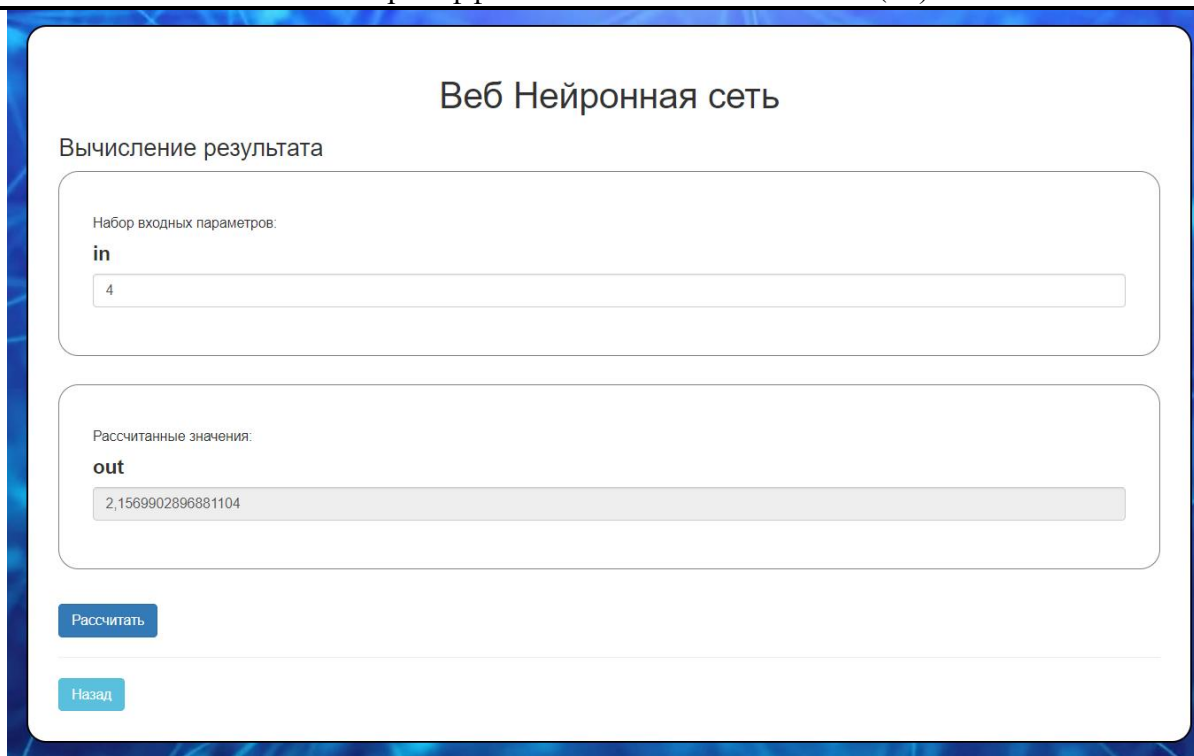


Рисунок 8 - Результат веб-программы

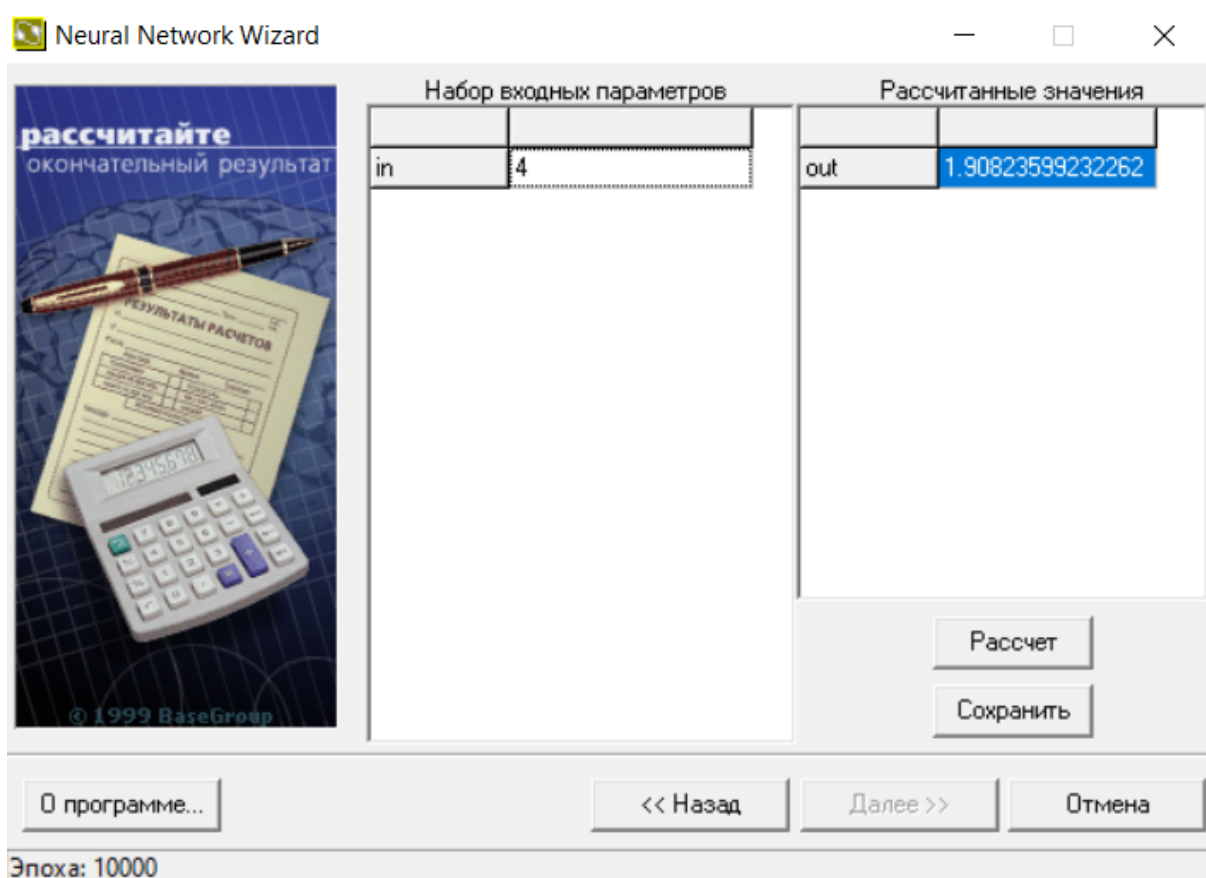


Рисунок 9 - Результат программы «Neural Network Wizard»

Таблица 2 – Результаты вычисления квадратного корня у веб-программы «ВНС» и «Neural Network Wizard»

Входные данные	Веб нейронная сеть (ВНС)	Neural Network Wizard	Погрешность Веб нейронная сеть (ВНС)	Погрешность Neural Network Wizard
1	0.9	1.2	-0.1	0.2
2	1.3	1.3	-0.1	-0.1
3	1.8	1.6	-0.1	0.2
4	2.1	1.9	0.1	-0.1
5	2.3	2.1	-0.1	0.1
6	2.4	2.4	0	0
7	2.5	2.6	-0.1	0
8	2.7	2.8	-0.1	0
9	3.2	3	0.2	0
10	3	3.2	-0.1	0.1
11	3.3	3.3	0	0
12	3.4	3.4	0	0
13	3.9	3.5	0.3	0.1
14	3.9	3.6	0.2	-0.1
15	3.9	3.7	0.1	-0.1
16	3.9	3.7	-0.1	-0.3

Данная веб-ориентированная программа «Веб нейронная сеть» (ВНС) предназначена для обучения студентов направления «Информационные системы и технологии». Также веб-программа «ВНС» будет улучшаться для другой области, например для распознавания рисунков, изображений и фотографий, а также будет улучшен алгоритм обучения нейронной сети, используя более правильный подход к библиотеке BrainJS.

Список литературы

1. Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов / Джон Дакетт; [пер. с англ. М. Lin H., Wang C. Influences of electromagnetic radiation distribution on chaotic dynamics of a neural network // Applied Mathematics and Computation, Т. 369, 2020, С. 124840
2. Wang X., Che M., Wei Y. Neural network approach for solving nonsingular multi-linear tensor systems // Journal of Computational and Applied Mathematics, In press, corrected proof, 2019, С. 112569
3. Taherkhani A., Belatreche A., Li Y., Cosma G., P.Maguiree L., McGinnityae T.M. A review of learning in biologically plausible spiking neural networks // Neural Networks, Т. 122, 2020, С. 253-272
4. Sahiner H., Liu X. Gamma spectral analysis by artificial neural network coupled with Monte Carlo simulations // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, In press, corrected proof, 2019, С. 163062

5. Heo S., Lee Jay. H. Parallel neural networks for improved nonlinear principal component analysis // *Computers & Chemical Engineering*, Т. 127, 2019, С. 1-10
6. Ayala D., Borrego A., Hernández I., Ruiz D. A neural network for semantic labelling of structured information // *Expert Systems with Applications*, Т. 143, 2020, С. 113053
7. Li T., Zhang Z., Chen H. Predicting the combustion state of rotary kilns using a Convolutional Recurrent Neural Network // *Journal of Process Control*, Т. 84, 2019, С. 207-214
8. Jiang C., Chen Q. Construction of blind restoration model for super-resolution image based on chaotic neural network // *Chaos, Solitons & Fractals*, In press, corrected proof, 2019, С. 109498
9. Li W., Li M., Qiao J., Guo X. A feature clustering-based adaptive modular neural network for nonlinear system modeling // *ISA Transactions*, In press, journal pre-proof, 2019
10. Patiño-Saucedo A., Rostro-Gonzalez H., Serrano-Gotarredona T., Linares-Barranco B. Event-driven implementation of deep spiking convolutional neural networks for supervised classification using the SpiNNaker neuromorphic platform // *Neural Networks*, Т. 121, 2020, С. 319-328
11. Choi A., Wang R., Darwiche A. On the relative expressiveness of Bayesian and neural networks // *International Journal of Approximate Reasoning*, Т. 113, 2019, С. 303-323
12. Dong Y., Fu Z., Peng Y., Zheng Y., Li X. Precision fertilization method of field crops based on the Wavelet-BP neural network in China // *Journal of Cleaner Production*, In press, corrected proof, 2019, С. 118735
13. Kobayashi M. Storage capacity of hyperbolic Hopfield neural networks // *Neurocomputing*, Т. 369, 5 2019, С. 185-190
14. Aguilar J. Z., Gómez-Aguilar J. F., Alvarado-Martínez V. M., Romero-Ugalde H. M. Fractional order neural networks for system identification // *Chaos, Solitons & Fractals*, Т. 130, 2020, С. 109444
15. Sideratos G., Ikononopoulos A., Hatziaargyriou N. D. A novel fuzzy-based ensemble model for load forecasting using hybrid deep neural networks // *Electric Power Systems Research*, Т. 178, 2020, С. 106025
16. BrainJS - Neural networks in JavaScript [Электронный ресурс] / GitHub – Режим доступа: <https://github.com/BrainJS/brain.js>, свободный. (Дата обращения 21.11.2019)
17. CSS-фреймворк Bootstrap [Электронный ресурс] / Bootstrap - Режим доступа: <https://getbootstrap.com/>, свободный. (Дата обращения 21.11.2019)
18. Нейронные сети для начинающих. Часть 1 [Электронный ресурс] / Хабр - Режим доступа: <https://habr.com/ru/post/312450/>, свободный (Дата обращения 21.11.2019)
19. Нейронные сети для начинающих. Часть 2 [Электронный ресурс] / Хабр - Режим доступа: <https://habr.com/ru/post/313216/>, свободный (Дата обращения 21.11.2019)
20. Как работает нейронная сеть: алгоритмы, обучение, функции активации, потери [Электронный ресурс] / Neurohive.io - Режим доступа: <https://neurohive.io/ru/osnovy-data-science/osnovy-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/>, свободный (Дата обращения 21.11.2019)
21. Программа Веб нейронная сеть [Электронный ресурс] / Web neuron network - Режим доступа: <http://kmc.pgusa.ru/neuron>, свободный (Дата обращения 21.11.2019)
22. Neuron Network Wizard скачать [Электронный ресурс] / FreeSoft - Режим доступа: https://freesoft.ru/windows/neural_network_wizard_v17, свободный (Дата обращения 21.11.2019)

23. Порядок работы с Neuron Network Wizard [Электронный ресурс] / Студопедия - Режим доступа: https://studopedia.ru/18_2723_poryadok-raboti-s-Neural-Network-Wizard.html, свободный (Дата обращения 21.11.2019)

References

1. Lin H., Wang C. Influences of electromagnetic radiation distribution on chaotic dynamics of a neural network // *Applied Mathematics and Computation*, T. 369, 2020, С. 124840
2. Wang X., Che M., Wei Y. Neural network approach for solving nonsingular multi-linear tensor systems // *Journal of Computational and Applied Mathematics*, In press, corrected proof, 2019, P. 112569
3. Taherkhani A., Belatreche A., Li Y., Cosma G., P.Maguiree L., McGinnityae T.M. A review of learning in biologically plausible spiking neural networks // *Neural Networks*, T. 122, 2020, P. 253-272
4. Sahiner H., Liu X. Gamma spectral analysis by artificial neural network coupled with Monte Carlo simulations // *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, In press, corrected proof, 2019, P.163062
5. Heo S., Lee Jay. H. Parallel neural networks for improved nonlinear principal component analysis // *Computers & Chemical Engineering*, T. 127, 2019, P. 1-10
6. Ayala D., Borrego A., Hernández I., Ruiz D. A neural network for semantic labelling of structured information // *Expert Systems with Applications*, T. 143, 2020, P. 113053
7. Li T., Zhang Z., Chen H. Predicting the combustion state of rotary kilns using a Convolutional Recurrent Neural Network // *Journal of Process Control*, T. 84, 2019, P. 207-214
8. Jiang C., Chen Q. Construction of blind restoration model for super-resolution image based on chaotic neural network // *Chaos, Solitons & Fractals*, In press, corrected proof, 2019, P. 109498
9. Li W., Li M., Qiao J., Guo X. A feature clustering-based adaptive modular neural network for nonlinear system modeling // *ISA Transactions*, In press, journal pre-proof, 2019
10. Patiño-Saucedo A., Rostro-Gonzalez H., Serrano-Gotarredona T., Linares-Barranco B. Event-driven implementation of deep spiking convolutional neural networks for supervised classification using the SpiNNaker neuromorphic platform // *Neural Networks*, T. 121, 2020, P. 319-328
11. Choi A., Wang R., Darwiche A. On the relative expressiveness of Bayesian and neural networks // *International Journal of Approximate Reasoning*, T. 113, 2019, P. 303-323
12. Dong Y., Fu Z., Peng Y., Zheng Y., Li X. Precision fertilization method of field crops based on the Wavelet-BP neural network in China // *Journal of Cleaner Production*, In press, corrected proof, 2019, P. 118735
13. Kobayashi M. Storage capacity of hyperbolic Hopfield neural networks // *Neurocomputing*, T. 369, 5 2019, P. 185-190
14. Aguilar J. Z., Gómez-Aguilar J. F., Alvarado-Martínez V. M., Romero-Ugalde H. M. Fractional order neural networks for system identification // *Chaos, Solitons & Fractals*, T. 130, 2020, P. 109444
15. Sideratos G., Ikonopoulou A., Hatziaargyriou N. D. A novel fuzzy-based ensemble model for load forecasting using hybrid deep neural networks // *Electric Power Systems Research*, T. 178, 2020, P. 106025

16. BrainJS - Neural networks in JavaScript, GitHub. [Online]. Available: <https://github.com/BrainJS/brain.js> [Accessed: 21-Nov-2019]
 17. CSS-framework Bootstrap, Bootstrap. [Online]. Available: <https://getbootstrap.com/> [Accessed: 21-Nov-2019]
 18. Neural networks for beginners. Part 1, Habr. [Online]. Available: <https://habr.com/ru/post/312450/> [Accessed: 21-Nov-2019]
 19. Neural networks for beginners. Part 2, Habr. [Online]. Available: <https://habr.com/ru/post/313216/> [Accessed: 21-Nov-2019]
 20. How a neural network works: algorithms, training, activation functions, losses, Neurohive.io. [Online]. Available: <https://neurohive.io/ru/osnovy-data-science/osnovy-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/> [Accessed: 21-Nov-2019]
 21. Program Web neuron network, Web neuron network. [Online]. Available: <http://kmc.pgusa.ru/neuron> [Accessed: 21-Nov-2019]
 22. Neuron Network Wizard download, FreeSoft. [Online]. Available: https://freesoft.ru/windows/neural_network_wizard_v17 [Accessed: 21-Nov-2019]
 23. The order of work with Neuron Network Wizard, Studopedia [Online]. Available: https://studopedia.ru/18_2723_poryadok-raboti-s-Neural-Network-Wizard.html [Accessed: 21-Nov-2019]
-