

## Использование нейронной сети в среде Scilab

*Шайдунов Александр Александрович*

*Приамурский государственный университет им. Шолом-Алейхема*

*студент*

### Аннотации

В данной работе рассмотрено пример использования нейронной сети с помощью средств среды Scilab. Рассмотрено несколько работ, связанных с нейронными сетями и работой в среде Scilab. Было проверено использование нейронной сети с разным числом данных для обработки. В результате было получены результаты, из которых следует, что при большем количестве данных для обработки, результат получается точнее. Данная работа может быть полезна для людей, которые работают с нейронными сетями или со средой Scilab. В работе присутствуют описания множества других работ, раскрывают сложности работы с нейронными сетями.

**Ключевые слова:** Исследование, Scilab, Data Analysis, Neural Network.

## Using a neural network in a Scilab environment

*Shaidurov Aleksandr Aleksandrovich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

In this paper, we consider an example of using a neural network using Scilab environment tools. Several works related to neural networks and work in the Scilab environment are considered. The use of a neural network with a different amount of data for processing was tested. As a result, results were obtained, from which it follows that with more data for processing, the result is more accurate. This work may be useful for people who work with neural networks or with the Scilab environment. The work contains descriptions of many other works, reveals the difficulties of working with neural networks.

**Keywords:** Research, Scilab, Data Analysis, Neural Network.

В настоящее время идёт активное развитие нейронных сетей. Сервисы и устройства, использующие нейронную сеть, набирают популярность. Нейронные сети позволяют упростить множество процессов, которые используют сложные алгоритмы или занимают много времени. Также позволяют воспроизводить сложные действия или задачи. Особенность нейросетей является, реализация различных алгоритмов вычисления при неполном или не точном описании заданной задачи. Обучаются нейросети за счёт нахождения коэффициентов или зависимости входных и выходных

данных. В сети интернет существует множество программного обеспечения для исследований и работы с нейронными сетями.

В работе С. Хио и Д. Ли представлена параллельная архитектура нейронной сети. Предлагаемая архитектура может быть получена путем систематического сокращения сетевых подключений полностью связанной автоассоциативной нейронной сети, что приводит к разделению нейронных сетей. В результате могут быть получены более независимые главные компоненты, чем при подходе одновременной экстракции. Предлагаемая в работе архитектура, может рассматриваться, как перестановка нескольких нейронных сетей для последовательного выделения в параллельной настройке, таким образом, обучение сети становится более эффективным. Имитация тематических исследований проводится для иллюстрации преимуществ предлагаемой архитектуры [1].

В статье А. Патињо-Соседоа, О. Ростро-Гонсалеса, Т. Серрано-Готарредонабы и Б. Линарес-Барранкоба показано, что на нейроморфном оборудовании возможна реализация современных моделей как для MNIST, так и для наборов данных распознавания цифр NMNIST на основе событий. Было использовано два подхода: путем непосредственного преобразования классической нейронной сети в ее заостренную версию и путем обучения зарождающейся сети с нуля. В обоих случаях были выполнены программные симуляции и реализации на машине SpiNNaker 103. С помощью этого метода предоставлены новое представление о возможностях, для эффективного кодирования пространственно-временной информации [2].

В работе М. Кобаяши анализируется емкость хранилища HHNN и сравнивается с емкостью CHNN. Комплексная нейронная сеть Хопфилда (CHNN) представляет собой многоуровневую модель нейронной сети Хопфилда и применяется для хранения многоуровневых данных. Гиперболическая нейронная сеть Хопфилда (HHNN) является еще одной многоуровневой моделью. Емкость хранилища является важной проблемой нейронных сетей Хопфилда [3].

В статье «Fractional order neural networks for system identification» представлено объединение двух подходов, чтобы использовать нейронную сеть дробного порядка (FONN) для идентификации системы. Нейронные сети и исчисление дробного порядка оказались мощными инструментами 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. При сравнении с другими подходами было показано, что предложенная модель FONN достигла большей точности с меньшим количеством параметров [4].

В работе «Neural network-based approaches for biomedical relation classification: A review» описан недавний прогресс нейросетевых подходов для классификации биомедицинских отношений. Были представлены общие основы для нейросетевых подходов в извлечении биомедицинских отношений и предварительно встроенных ресурсах встраивания слов. Были

обсуждены подходы на основе нейронных сетей, включая сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN). В заключение мы описываем оставшиеся проблемы и намечаем будущие направления [5].

В статье Д.С. Сохала особое внимание уделяется разработке системы распознавания символов с использованием SciLab, бесплатного программного обеспечения с открытым исходным кодом и является наиболее многообещающей альтернативой MatLab. В предлагаемой работе разработана система распознавания символов для извлечения печатного текста из изображения с использованием поисковой системы на основе самоорганизующихся карт Кохенена. SOM, будучи неконтролируемым методом обучения, обладает превосходными свойствами извлечения. Используются образцы одинаковых символов, которые ориентированы под одним и тем же углом, но с разным размером, цветом и шрифтами. Благодаря самоорганизующейся карте вместе с алгоритмом кластеризации «К», означает использование программного обеспечения SciLab, система достигла замечательной точности от 99% до 100% при тестировании для различных изображений ввода текста [6].

В работе «Compiling Scilab to high performance embedded multicore systems» описывается проблема, которая заключается в выражении параллелизма с чисто императивным языком программирования, которым обычно является С. Этот традиционный подход ограничивает отображение, разбиение и генерацию оптимизированного параллельного кода, и, следовательно, достижимую производительность и энергопотребление приложений из разных областей. Архитектурно ориентированная параллелизация для высокопроизводительных встраиваемых многоядерных систем, использующих европейский проект Scilab (ALMA), направлена на преодоление этих препятствий посредством введения и эксплуатации набора инструментов на основе Scilab, который позволяет эффективно отображать приложения на многопроцессорных платформах с высоким уровнем абстракции. Целостное решение цепочки инструментов ALMA позволяет скрыть сложность как приложения, так и архитектуры, что приводит к лучшему восприятию, снижению стоимости разработки и сокращению времени выхода на рынок. В связи с технологическими ограничениями в разработке микросхем, окончанием экспоненциального роста тактовых частот и неизбежным повышением требований к производительности вычислений, ALMA является фундаментальным шагом вперед в необходимом внедрении новых вычислительных парадигм и методологий [7].

В работе Ч. Цайяя предоставлена альтернатива, бесплатную Scilab, с помощью которой можно создавать пользовательские графические интерфейсы и распространять их без дополнительных ограничений и затрат. Scilab разработал постпроцессор для визуализации данных анализа выгорания топливного стержня, рассчитанных FRAPCON-3.4. Этот постпроцессор включает графический интерфейс пользователя, предоставляя пользователям быстрый обзор характеристик численных результатов с помощью двумерных и трехмерных графиков, а также анимации

распределения температуры топлива. Входной файл оценочного случая, предоставленный группой пользователей FRAPCON, был применен для демонстрации построения постпроцессора с графическим интерфейсом пользователя с помощью объектно-ориентированного инструмента графического интерфейса пользователя, а также возможностей визуализации в Scilab [8].

В работе Г. Йена было продемонстрировано, что глубокие нейронные сети превосходят классические алгоритмы машинного обучения при решении реальных задач. Тем не менее, самые успешные глубокие нейронные сети были созданы вручную с нуля, принимая во внимание знание проблемной области. Этот подход часто требует очень значительных временных и вычислительных ресурсов. В этой работе предлагается новый алгоритм, основанный на оптимизации роя частиц (PSO), способный к быстрой сходимости по сравнению с другими эволюционными подходами, для автоматического поиска значимых архитектур глубоких сверточных нейронных сетей (CNN) для задач классификации изображений, названный psoCNN. Была разработана новая стратегия прямого кодирования и оператор скорости, позволяющий оптимизировать использование PSO с CNN. Экспериментальные результаты показывают, что psoCNN может быстро найти хорошие архитектуры CNN, которые достигают качественных характеристик, сравнимых с современными проектами [9].

В статье Г. Чжанга, Ч. Зенга и Д. Нинга представлен класс зависящих от состояния коммутируемых нейронных сетей с инерционными элементами и распределенными задержками. Несколько новых результатов получены для обеспечения экспоненциальной синхронизации таких коммутируемых нейронных сетей с использованием новой гибридной схемы управления и теории устойчивости Ляпунова. Наконец, моделирование дается, чтобы показать достоверность полученных результатов. Новый полезный метод управления из этой статьи расширяет область применения коммутируемых нейронных сетей [10].

Цель данного исследования — используя доступное программное обеспечение улучшить результаты, полученные после использования нейронной сети.

Сравнение результатов, полученных после использования нейронной сети с разным числом входных и выходных данных. В качестве программного обеспечения используется SciLab — пакет прикладных математических программ, предоставляющий открытое окружение для инженерных и научных расчётов. Используя среду Scilab, можно создать собственную нейросеть. Данная среда является доступным аналогом среды Matlab. Данная среда постоянно развивается и совершенствуется. В системе присутствуют такие возможности и инструменты, как :2D и 3D графики, анимация, линейная алгебра, разреженные матрицы, полиномиальные и рациональные функции, интерполяция, аппроксимация, симуляция: решение ОДУ и ДУ, scicos — это гибридная система моделирования динамических систем и симуляции, дифференциальные и не дифференциальные

оптимизации, обработка сигналов, параллельная работа, статистика, работа с компьютерной алгеброй. О том как использовать среду в маш можно в научных работах [6,7,8,11].

Для начала работы необходимо установить SciLab версии 5.5.2, далее загрузить и установить необходимые модули. Для этого необходимо во вкладке «Инструменты» выбрать пункт «Управление модулями ATOMS».

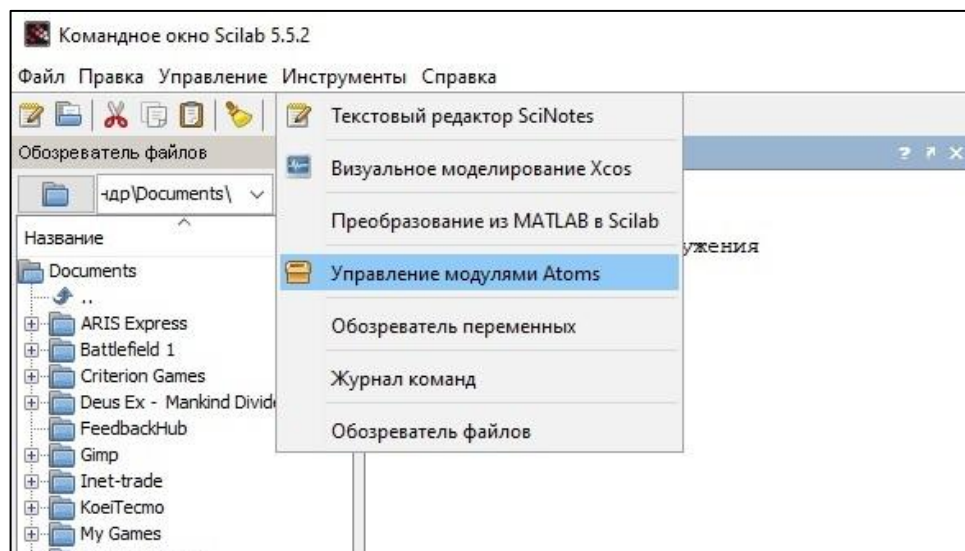


Рис.1. Управление модулями Atoms

Далее необходимо выбрать в папке «Data Analysis» модули ANN Toolbox и Neural Network Module и выбрать пункт «установить». Данные модули предназначены для работы с нейронными сетями.

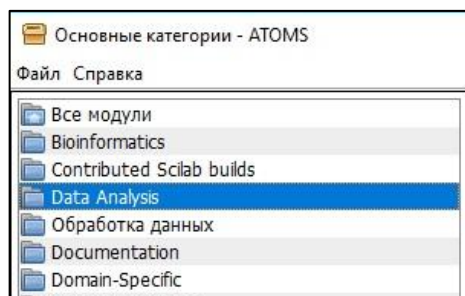


Рис.2. Папка Data Analysis

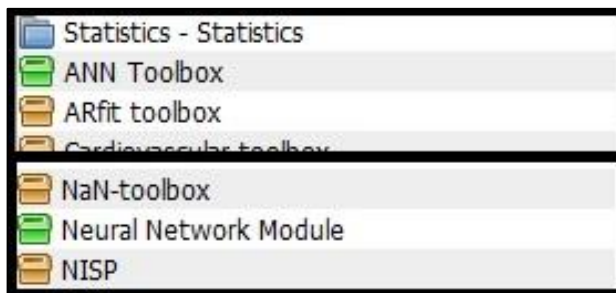


Рис.3. Необходимые расширения

После загрузке необходимо перезагрузить программу. При новом запуске программы в командном окне будет выведена информация о загружена информация об установке модуля.

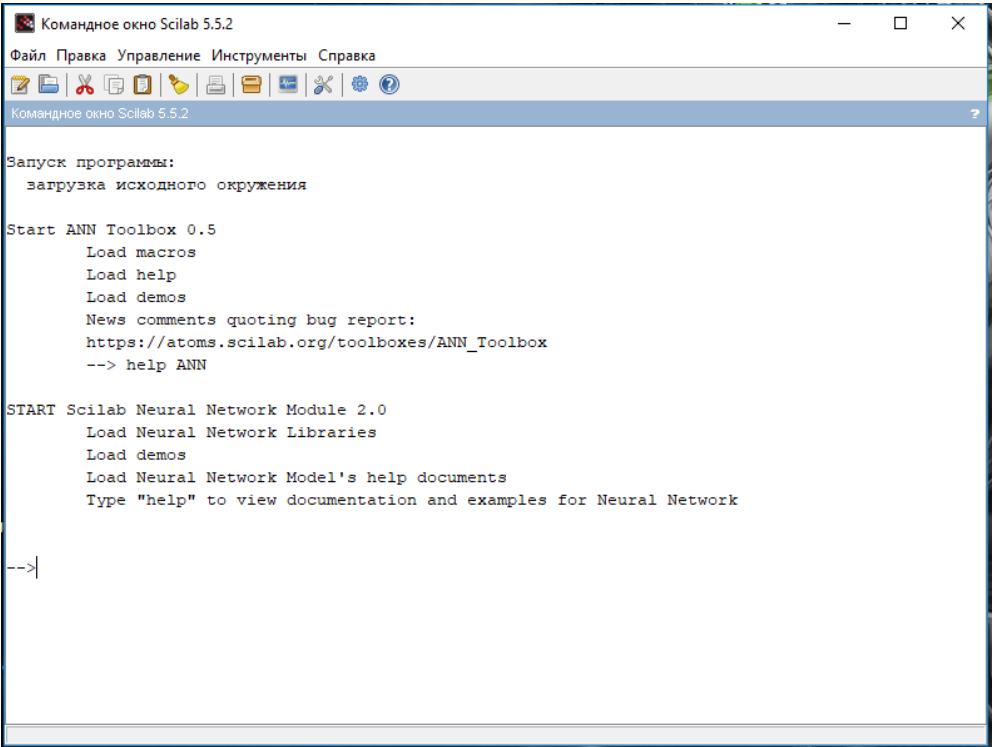


Рис.4.Информация о успешной загрузке модулей

В редактор SciNotes вносится весь необходимые код программы. В данной работе он включает в себя код программы и данные для анализа сетью.

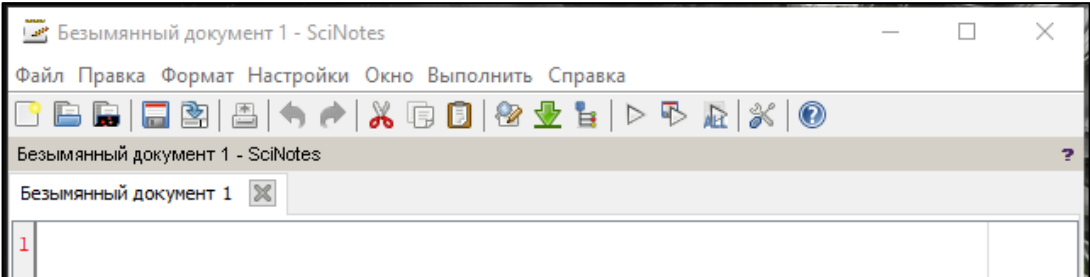


Рис.5. Текстовый редактор SciNotes

Далее, необходимо добавить данные, для обучения нейронной сети. Для этого были искусственно созданы и внесены данные.

|    |   |      |   |   |   |   |   |   |    |       |   |      |    |       |     |   |    |    |    |
|----|---|------|---|---|---|---|---|---|----|-------|---|------|----|-------|-----|---|----|----|----|
| X1 | 0 | 0    | 1 | 1 | 2 | 2 | 3 | 3 | 3  | 3     | 4 | 4    | 4  | 4     | 5   | 5 | 5  | 5  | 5  |
| X2 | 0 | 0,01 | 1 | 3 | 5 | 7 | 3 | 6 | 8  | 8,15  | 2 | 5,11 | 7  | 9,05  | 1,1 | 3 | 5  | 7  | 8  |
| y  | 0 | 0,01 | 2 | 4 | 7 | 9 | 6 | 9 | 11 | 11,15 | 6 | 9,11 | 11 | 13,05 | 6,1 | 8 | 10 | 12 | 13 |

|    |   |       |    |    |      |    |    |    |    |    |    |    |      |       |    |    |    |       |       |
|----|---|-------|----|----|------|----|----|----|----|----|----|----|------|-------|----|----|----|-------|-------|
| X1 | 6 | 6,03  | 6  | 6  | 7,35 | 7  | 7  | 7  | 7  | 8  | 8  | 8  | 9    | 9     | 9  | 9  | 9  | 9     | 9,01  |
| X2 | 2 | 4     | 7  | 9  | 2,01 | 5  | 6  | 7  | 9  | 3  | 5  | 7  | 3,5  | 4,01  | 5  | 7  | 8  | 8,06  | 9     |
| y  | 8 | 10,03 | 13 | 15 | 9,36 | 12 | 13 | 14 | 16 | 11 | 13 | 15 | 12,5 | 13,01 | 14 | 16 | 17 | 17,06 | 18,01 |

Рис.6. Данные для анализа

Входные параметры «Р» будут значения «X1» и «X2» из таблицы «Данные сложения двух чисел», а целевые параметры «Т» значения «Y». Данные будут использованы для обучения нейронной сети.

Далее для обучения сети используем функцию «ann\_FFBP\_gd» с алгоритмом обратного распространения ошибки обучения.

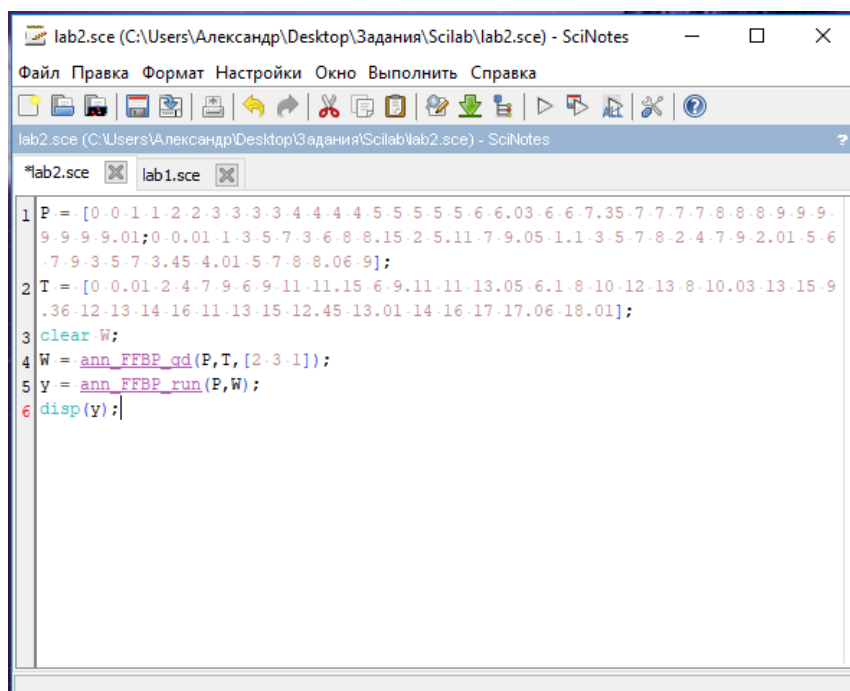
$W = \text{ann\_FFBP\_gd}(P, T, N, af, lr, \text{itermax}, \text{mse\_min}, \text{gd\_min})$

Где:

- Р — входные параметры;
- Т — целевые параметры;
- N — число нейронов в каждом слое, включая входной и выходной слой;
- af — активация функции от 1-го скрытого слоя к выходному слою;
- lr — скорость обучения;
- itermax — максимальное эпоха для тренировок;
- mse\_min — минимальная ошибка;
- gd\_min — минимальный градиент для подготовки к остановке.

Для демонстрации работы, достаточно использовать только первые 3 значений, другие значения можно оставляем пустыми, они будут заменены по умолчанию.

Для использования обученной нейронной сети будет использована функция «ann\_FFBP\_run».



```

lab2.sce (C:\Users\Александр\Desktop\Задания\Scilab\lab2.sce) - SciNotes
Файл  Правка  Формат  Настройки  Окно  Выполнить  Справка
lab2.sce (C:\Users\Александр\Desktop\Задания\Scilab\lab2.sce) - SciNotes
*lab2.sce  lab1.sce
1 P = [0 0 1 1 2 2 3 3 3 3 4 4 4 4 5 5 5 5 5 6 6 03 6 6 7 35 7 7 7 7 8 8 8 9 9 9 9
9 9 9 9 01; 0 0 01 1 3 5 7 3 6 8 8 15 2 5 11 7 9 05 1 1 3 5 7 8 2 4 7 9 2 01 5 6
7 9 3 5 7 3 45 4 01 5 7 8 8 06 9];
2 T = [0 0 01 2 4 7 9 6 9 11 11 15 6 9 11 11 13 05 6 1 8 10 12 13 8 10 03 13 15 9
36 12 13 14 16 11 13 15 12 45 13 01 14 16 17 17 06 18 01];
3 clear W;
4 W = ann_FFBP_gd(P,T,[2 3 1]);
5 y = ann_FFBP_run(P,W);
6 disp(y);
  
```

Рис.7. Код для использования нейронной сети

При выборе пункта Выполнить. Откроется окно «Neural Network Training», в нём сообщается, что обучение длится 1000 итераций и среднеквадратичную ошибку.

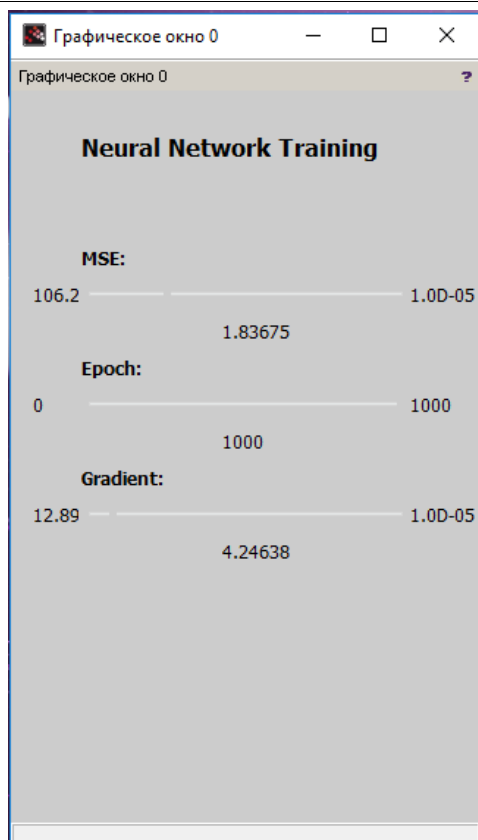


Рис.8. Окно Neural Network Training

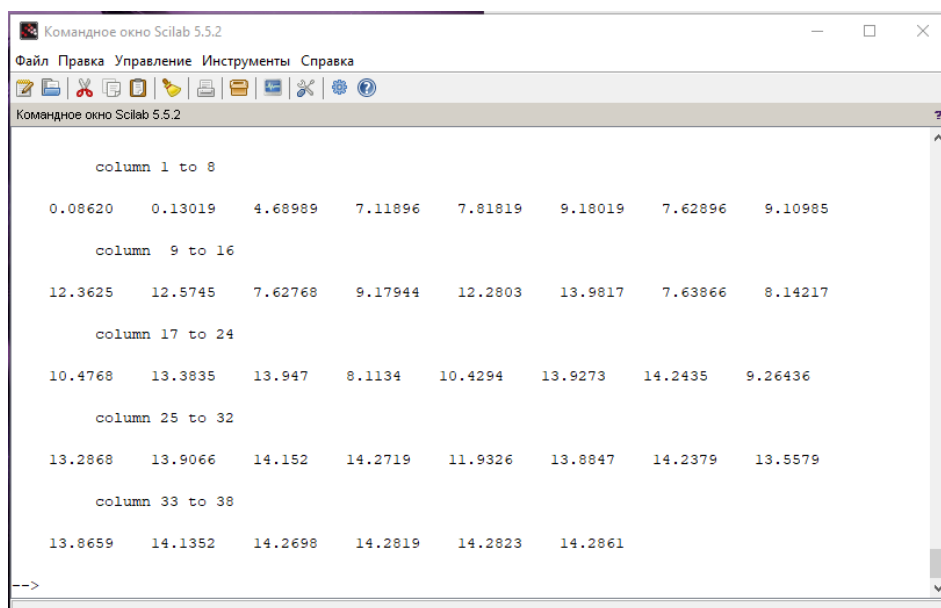


Рис.9. Результат обучения нейронной сети в Scilab

Чтобы использовать нейронную сеть, необходима функция «ann\_FFBP\_run». Далее её можно использовать на примере суммы двух чисел.

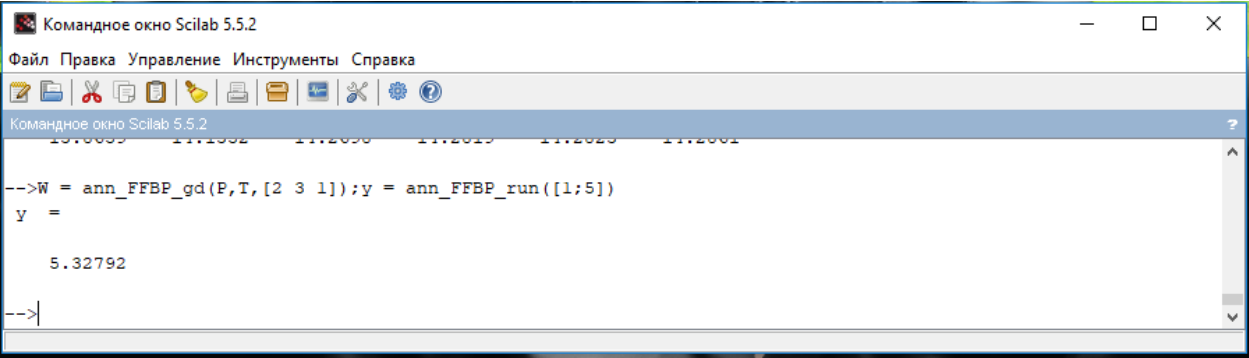


Рис.10. Первый результат

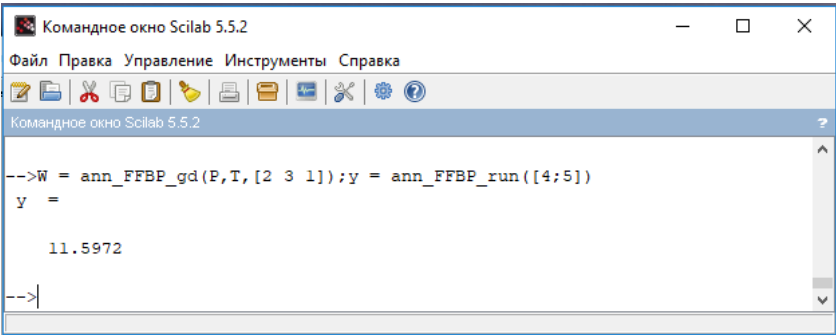


Рис.11.Второй результат

В итоге, в командном окне был выведен результат суммы двух чисел. Далее таблица с данными была расширена. Для обработки была расширена база введенных в нейронную сеть данных.

|    |   |      |     |   |      |   |   |   |     |    |   |   |   |   |   |    |    |   |   |   |
|----|---|------|-----|---|------|---|---|---|-----|----|---|---|---|---|---|----|----|---|---|---|
| X1 | 0 | 0    | 0   | 1 | 1    | 1 | 1 | 1 | 1   | 1  | 2 | 2 | 2 | 2 | 2 | 3  | 3  | 3 | 3 | 3 |
| X2 | 0 | 0,01 | 0,5 | 1 | 2,01 | 3 | 5 | 7 | 8,5 | 9  | 1 | 3 | 5 | 6 | 7 | 8  | 9  | 3 | 4 | 5 |
| y  | 0 | 0,01 | 0,5 | 2 | 3,01 | 4 | 6 | 8 | 9,5 | 10 | 3 | 5 | 7 | 8 | 9 | 11 | 12 | 6 | 7 | 8 |

|    |   |    |    |      |    |   |   |      |    |      |    |      |     |   |    |    |    |       |    |    |
|----|---|----|----|------|----|---|---|------|----|------|----|------|-----|---|----|----|----|-------|----|----|
| X1 | 3 | 3  | 3  | 3    | 3  | 4 | 4 | 4    | 4  | 4    | 4  | 4    | 5   | 5 | 5  | 5  | 5  | 5     | 5  | 5  |
| X2 | 6 | 7  | 8  | 8,15 | 9  | 2 | 4 | 5,11 | 6  | 6,5  | 7  | 9,05 | 1,1 | 3 | 5  | 6  | 7  | 7,78  | 8  | 9  |
| y  | 9 | 10 | 11 | 11,2 | 12 | 6 | 8 | 9,11 | 10 | 10,5 | 11 | 13,1 | 6,1 | 8 | 10 | 11 | 12 | 12,78 | 13 | 14 |

|    |   |   |    |    |    |    |    |   |      |    |    |    |    |       |    |    |    |   |    |    |
|----|---|---|----|----|----|----|----|---|------|----|----|----|----|-------|----|----|----|---|----|----|
| X1 | 6 | 6 | 6  | 6  | 6  | 6  | 6  | 7 | 7,35 | 7  | 7  | 7  | 7  | 7,05  | 7  | 7  | 7  | 8 | 8  | 8  |
| X2 | 2 | 3 | 4  | 5  | 6  | 7  | 9  | 0 | 2,01 | 3  | 4  | 5  | 6  | 6,67  | 7  | 8  | 9  | 1 | 2  | 3  |
| y  | 8 | 9 | 10 | 11 | 12 | 13 | 15 | 7 | 9,36 | 10 | 11 | 12 | 13 | 13,72 | 14 | 15 | 16 | 9 | 10 | 11 |

|    |    |    |     |    |    |       |    |   |    |    |      |      |      |    |      |    |     |    |       |       |
|----|----|----|-----|----|----|-------|----|---|----|----|------|------|------|----|------|----|-----|----|-------|-------|
| X1 | 8  | 8  | 8   | 8  | 8  | 8,45  | 8  | 9 | 9  | 9  | 9    | 9    | 9    | 9  | 9    | 9  | 9   | 9  | 9     | 9,01  |
| X2 | 5  | 6  | 6,7 | 7  | 8  | 8,36  | 9  | 0 | 1  | 2  | 3,45 | 4,01 | 4,88 | 5  | 5,01 | 7  | 7,5 | 8  | 8,06  | 9     |
| y  | 13 | 14 | 15  | 15 | 16 | 16,81 | 17 | 9 | 10 | 11 | 12,5 | 13   | 13,9 | 14 | 14   | 16 | 16  | 17 | 17,06 | 18,01 |

Рис.12.Расширенные данные для обработки

Таким же образом был задан запрос, и получен результат.

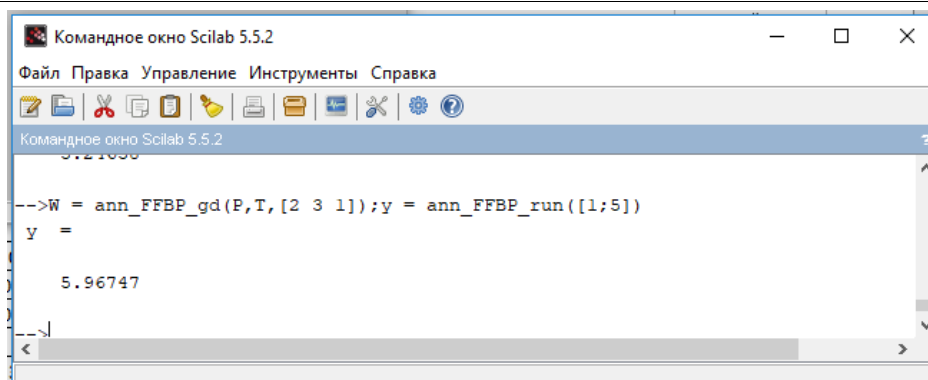


Рис.13.Результат работы нейронной сети

Также был зада дополнительный запрос.

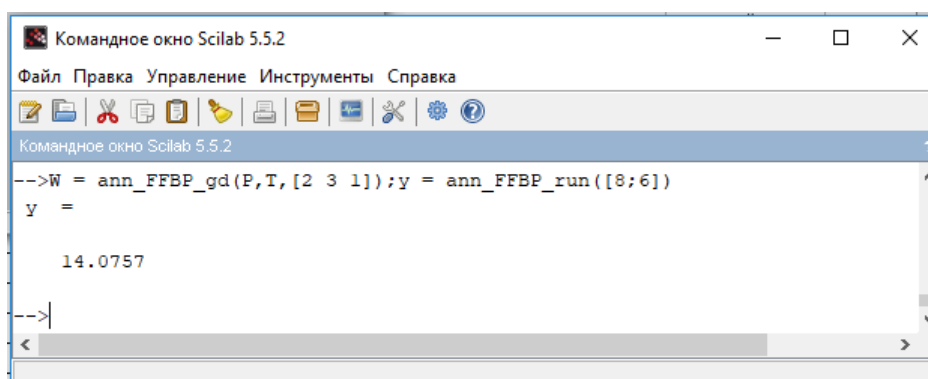


Рис.14. Результат работы нейронной сети

При втором вычислении полученный ответ имеет меньший разрыв от необходимого результата. Таким образом, при большем количестве входных данных, полученный результат является более точным.

Данная работа может помочь пользователям, которые хотят вести работы в среде SciLab или работать с нейронными сетями. Рассмотренные источники также могут оказаться полезными, как при расчётах с нейронными сетями, так и при работе в SciLab. Так в сравнении с работай У.С. Журавлёвы и Р.И. Баженова были улучшены результаты вычисления, результаты. Полученные результаты являются болие точными [11]. Работа Семенова Т.И., Загвоздкина А.В. и Загвоздкин В.А. позволяет оценить точность и возможности среды SciLab. Эта работы доказывает, что среда является хорошей альтернативой для использования при сложных посчётах [12]. По созданию нейронной сети в ближайшем аналоге Matlab можно найти в работе М.А. Афанасьевой [13].

Результатом работы стали результаты, полученные путём исследования, из которых следует, что чем больше данных для анализа в нейронной сети, тем точнее будет полученный результат.

**Библиографический список**

1. Heo S., Lee J.H. Parallel neural networks for improved nonlinear principal component analysis / Computers & Chemical Engineering, №127, 2019, Pages 1-10.
2. Patiño-Saucedo A., Rostro-Gonzalez H., Serrano-Gotarredona T., Linares-Barrancob B., Patiño-Saucedoa A., Rostro-Gonzaleza H., Serrano-Gotarredonab T., Linares-Barranco B. Event-driven implementation of deep spiking convolutional neural networks for supervised classification using the SpiNNaker neuromorphic platform / Neural Networks, №121, 2019, Pages 319-328.
3. Kobayashi M. Storage capacity of hyperbolic Hopfield neural networks / Neurocomputing, 2019.
4. Zuñiga C.J., Gómez-Aguilar J.F., Alvarado-Martínez V.M., Romero-Ugalde H.M. Fractional order neural networks for system identification / Chaos, Solitons & Fractals №130, 2019.
5. Zhang Y., Lin H., Yang Z., Wang J., Sun Y., Xu B., Zhao Z. Neural network-based approaches for biomedical relation classification: A review / Journal of Biomedical Informatics Volume 99, 2019.
6. J.S.Sohal Improvement of artificial neural network based character recognition system, using SciLab / Optik, №127, Issue 22, November 2016, Pages 10510-10518.
7. Stripf T., Oey O., Bruckschloegl T., Becker J., Rauwerda G., Sunesen K., Goulas G., PanayiotisAlefragis, N.S. Vorosc, Derriend S., Sentieysd O., Kavvadias N., Dimitroulakis G., Masselos K., Kritharidis D., Mitas N., Perschke T. Compiling Scilab to high performance embedded multicore systems / Microprocessors and Microsystems, №37, 2013, Pages 1033-1049.
8. Tsai C.W. Visualization of fuel rod burnup analysis by Scilab / Nuclear Engineering and Design, № 265, December 2013, Pages 341-347.
9. Yen G.G. Particle swarm optimization of deep neural networks architectures for image classification / Swarm and Evolutionary Computation, №49, September 2019, Pages 62-74.
10. Zhang G., Zeng Z, Ning D. Novel results on synchronization for a class of switched inertial neural networks with distributed delays / Information Sciences, №511, 2020, Pages 114-126.
11. Журавлёв У.С. и Баженов Р.И. Нейронные сети в Scilab / Постулат, №1, 2017.
12. Семенова Т.И., Загвоздкина А.В., Загвоздкин В.А. изучение численных методов с использованием средств пакета SciLab / Экономика и качество систем связи, №4, 2017, с.60-69.
13. Афанасьевой М.А. Создание и обучение нейронных сетей в системе Matlab / Молодой ученый, №4, 2014, 85-88 с.