

УДК 004.92: 004.032.26

DOI 10.51691/2541-8327\_2023\_8\_1

***РАЗРАБОТКА JAVASCRIPT-БИБЛИОТЕКИ «NNDRAWABLE» ДЛЯ  
ИЗОБРАЖЕНИЯ НЕЙРОННЫХ СЕТЕЙ В WEB***

***Вильданов А.Н.***

*к.ф.-м.н., доцент*

*ФГБОУ ВО «Уфимский университет науки и технологий», Нефтекамский филиал*

*Нефтекамск, Россия*

**Аннотация:** В работе представлена JavaScript-библиотека nndrawable, которая предоставляет программный интерфейс для рисования нейронных сетей в браузере. Данный интерфейс может пригодиться для визуального представления нейронных сетей, а также может быть полезен для понимания и работы с ними. Нейронные сети представляют собой сложные математические модели с множеством взаимосвязанных слоев и узлов, поэтому их визуализация может упростить понимание того, как они структурированы и как обрабатывают информацию. Сегодня canvas и SVG – популярные веб-технологии, используемые для создания графики и интерактивных визуализаций, но у них есть некоторые ключевые различия в их подходе и возможностях: canvas — это растровая технология, что означает, что он отображает графику, рисуя пиксели на растровом изображении, а SVG — это векторная технология, которая отображает графику с использованием математических уравнений для рисования фигур и линий. Предлагаемая библиотека nndrawable.js способна изображать нейронную сеть как с помощью canvas, так и с SVG. Библиотеку легко можно встроить и использовать на

любом сайте для изображения нейронных сетей, как в виде растровой, так и векторной графики. С кодом приложения можно ознакомиться на Github.

**Ключевые слова:** html, css, javascript, canvas, svg, нейронная сеть, полносвязный слой, библиотека, nndrawable, github.

***DEVELOPMENT OF A JAVASCRIPT LIBRARY «NNDRAWABLE» FOR  
DISPLAYING NEURAL NETWORKS IN THE WEB***

***Vildanov A.N.***

*candidate of Physical and Mathematical Sciences,*

*Ufa University of Science and Technology, Neftekamsk branch*

*Neftekamsk, Russia*

**Abstract:** The paper presents the nndrawable JavaScript library, which provides a programming interface for drawing neural networks in the browser. This interface can be useful for visual representation of neural networks, and can also be useful for understanding and working with them. Neural networks are complex mathematical models with many interconnected layers and nodes, so visualizing them can make it easier to understand how they are structured and how they process information. Today, canvas and SVG are popular web technologies used to create graphics and interactive visualizations, but they have some key differences in their approach and capabilities: canvas is a raster technology, which means that it renders graphics by drawing pixels on a bitmap, while SVG is a vector technology that renders graphics using mathematical equations to draw shapes and lines. The proposed nndrawable.js library is capable of drawing a neural network using both canvas and SVG. The library can be easily embedded and used on any site to display neural networks, both in the form of raster and vector graphics. The application code can be found on Github.

**Keywords:** html, css, javascript, canvas, svg, neural network, fully connected layer, library, nndrawable, github.

## Введение

Нейронные сети представляют собой сложные математические модели с множеством взаимосвязанных слоев и узлов, и их визуализация может упростить понимание того, как они структурированы и как обрабатывают информацию. Цель данного исследования – раскрыть возможности разработанной автором библиотеки на языке JavaScript, которая способна предоставить пользователю простой и удобный программный интерфейс для рисования нейронных сетей в браузере.

Предлагаемая библиотека `nndrawable.js` может изображать нейронную сеть как с помощью `canvas`, так и с `SVG`. Холст `HTML5 (canvas)` – это элемент `HTML5`, который позволяет динамически рисовать растровую графику и управлять ею с помощью `JavaScript` [1]. Элемент холста имеет прямоугольную форму и определенную ширину и высоту, которые можно указать с помощью атрибутов «ширина» и «высота». Такие методы, как «`fillRect()`», «`strokeRect()`» и «`drawImage()`» можно использовать для рисования фигур, линий и изображений на холсте.

Напротив, элемент `SVG (Scalable Vector Graphics)` представляет собой формат векторной графики на основе `XML`, который позволяет создавать графику любого размера без ущерба для качества. Графика `SVG` создается с помощью математических описаний форм и путей, которые интерпретируются средством визуализации `SVG`. Графику `SVG` также можно стилизовать и анимировать с помощью `CSS` и `JavaScript`.

Хотя элемент `Canvas` и элемент `SVG` могут использоваться для создания графики в `HTML5`, они имеют разные области применения и свойства. Элемент `Canvas` идеально подходит для создания сложной растровой графики, такой как Дневник науки | [www.dnevniknauki.ru](http://www.dnevniknauki.ru) | СМИ ЭЛ № ФС 77-68405 ISSN 2541-8327

игровая графика, имеет массу преимуществ [2], а элемент SVG лучше подходит для создания масштабируемой и интерактивной векторной графики, такой как диаграммы или инфографика.

### Описание возможностей библиотеки `nndrawable.js`

Чтобы подключить библиотеку `nndrawable.js` к веб-странице, нужно выполнить несколько простых шагов [3]:

- внутри тега `<head>` вашей веб-страницы создать тег `<script>`.
- внутри тега `<script>` указать путь к файлу библиотеки с помощью атрибута `src`:

```
<script type="text/javascript" src="nndrawable.js"></script>
```

На странице нужно подготовить текстовый блок для размещения в нем изображения нейронной сети:

```
<div id="draw_nn1"></div>
```

После этого можно создать объект – нейронную сеть, и слои для нее. Основным контейнером, содержащим все необходимые классы, является объект `nndraw` [4]. Для создания нейронной сети в библиотеке служит класс `NN`, который разработан с помощью объявления прототипа [5]:

```
let nn1 = new nndraw.NN();
```

Как известно, нейронная сеть состоит из слоев нейронов, которые связаны друг с другом. Каждый уровень получает входные данные от предыдущего уровня и передает выходные данные следующему уровню. Первый слой называется входным слоем, последний слой называется выходным слоем, а слои между ними называются скрытыми слоями.

Каждый нейрон в слое принимает линейную комбинацию входных данных, а затем применяет функцию активации к результату для получения выходных данных. Тип функции активации может варьироваться в зависимости от варианта использования.

Обучение нейронной сети включает в себя настройку весов и пороговых значений нейронов для достижения желаемого результата с учетом входных данных. Это делается с помощью метода оптимизации, такого как обратное распространение.

В зависимости от структуры и применения нейронные сети могут выглядеть очень по-разному. Некоторые архитектуры нейронных сетей включают сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN) и сети глубокого доверия (DBN).

Изобразить мы будем полносвязную нейронную сеть [6], в которой любой нейрон каждого слоя связан со всеми нейронами соседних слоев. Создадим и добавим слои в нейронную сеть [4]:

```
let layer1 = new nndraw.Layer(7); nn1.add(layer1);  
let layer2 = new nndraw.Layer(5); nn1.add(layer2);  
let layer3 = new nndraw.Layer(4); nn1.add(layer3);  
let layer4 = new nndraw.Layer(1); nn1.add(layer4);
```

Такой способ объявления, очевидно, означает, что у нашей нейронной сети будет четыре слоя. В первом слое – семь нейронов, далее пять, четыре и один нейрон соответственно (рисунок 1).

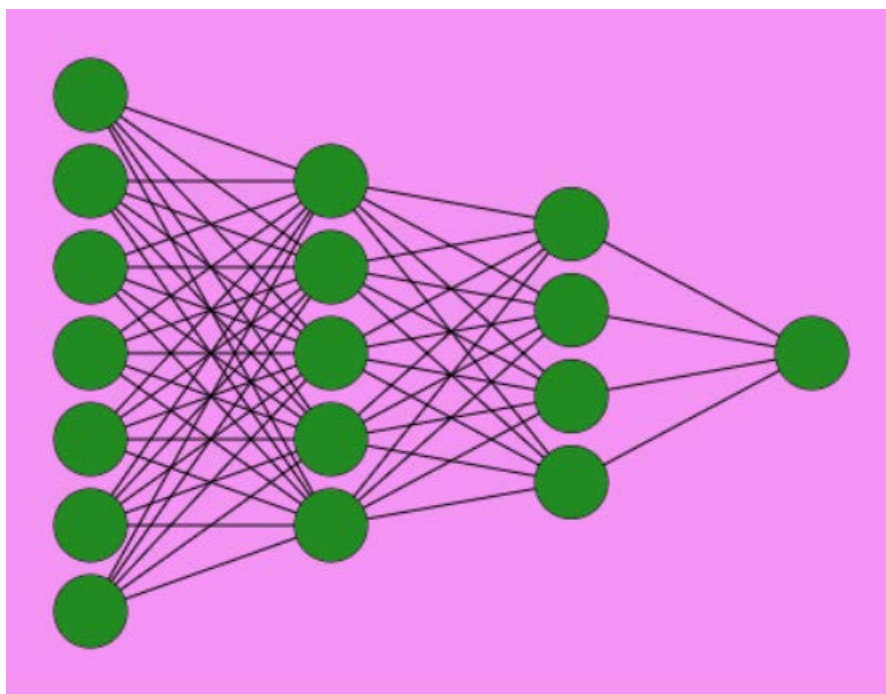


Рисунок 1 – Изображение нейронной сети

Теперь, когда задана модель нейронной сети, ее можно изобразить. Объявим переменную `div1` для подготовленного выше текстового блока:

```
var div1 = document.getElementById('draw_nn1');
```

и вызываем метод рисования нейронной сети внутри данного блока:

```
nn1.draw(div1);
```

Изображение нейронной сети построено (рисунок 1).

При рисовании нейронной сети можно указать и более тонкие настройки модели нейронной сети. Предусмотрены как глобальные настройки – для всей нейронной сети, так и настройки для каждого слоя. Например, изобразим модель нейронной сети, указав:

- цвет заливки общего фона;
- расстояние между слоями;
- величину радиуса круга, схематично изображающего нейрон;
- толщину линий;

- цвет линий.

Для этого достаточно написать следующий скрипт:

```
let nn2 = new nndraw.NN();  
nn2.canvasFillColor = "#FFD5DA";  
nn2.rx = 60;  
nn2.radius = 20;  
nn2.lineWidth = 2;  
nn2.strokeStyleColor = "#088F8F";
```

Далее объявляем слои нейронной сети, причем можно для каждого слоя отдельно задать, в свою очередь, следующие атрибуты:

- форму нейрона (прямоугольник или круг); иногда входные нейроны изображают в виде прямоугольника;
- количество нейронов в слое;
- цвет границы нейрона;
- цвет заливки нейрона.

Напишем следующий код для задания нейронной сети с пятью слоями и восемью входными данными (рисунок 2, а):

```
let layer = new nndraw.Layer( 8, "rectangle", "black",  
"#DAA520"); nn2.add(layer);  
layer = new nndraw.Layer(6, "circle", "black", "#228B22");  
nn2.add(layer);  
layer = new nndraw.Layer(5, "circle", "black", "#DAA520");  
nn2.add(layer);  
layer = new nndraw.Layer(3, "circle", "black", "#B0E0E6");  
nn2.add(layer);
```

```
layer = new nndraw.Layer(1, "circle", "black", "#D8BFD8");
nn2.add(layer);
```

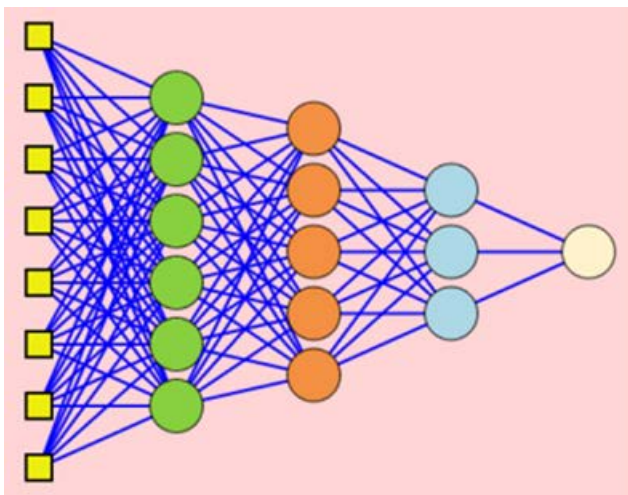


Рисунок 2, а – Изображение нейронной сети с входным слоем в canvas

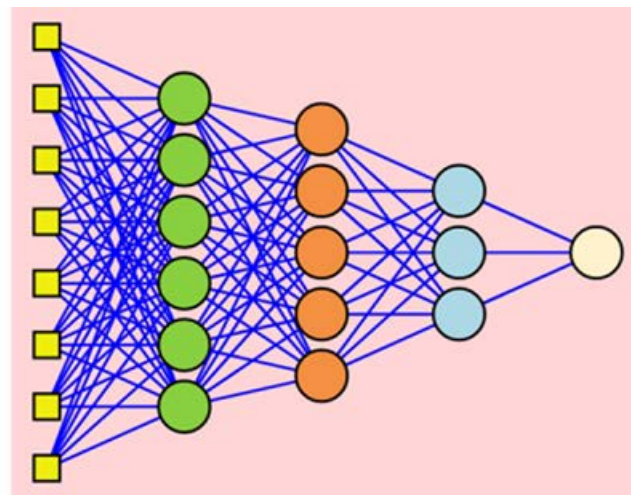


Рисунок 2, б – Изображение нейронной сети в SVG

Для рисования нейронной сети снова укажем новый текстовый раздел (блок), где будет нарисована нейронная сеть. В качестве второго параметра можно указать способ изображения: на основе canvas (по умолчанию), или на основе SVG:

```
var div2 = document.getElementById('draw_nn2');
nn2.draw(div2, 'canvas');
```

Изображение нейронной сети построено (рисунок 2, а). Наконец, как было обещано в начале, ту же нейронную сеть можно нарисовать с использованием векторной графики. Достаточно указать в методе `draw` другой параметр:

```
var div3 = document.getElementById('draw_nn3');
```



```
nn2.draw(div3, 'svg');
```

Качество изображения при этом лучше (рисунок 2, б).

Например, структуру нейронной сети для прогнозирования распределения диоксида азота в атмосферном воздухе города Таганрога [7], можно представить в виде рисунка 3.

В целом, наверное, будет правильно сказать, что программирование для canvas проще, чем для SVG, потому что canvas предоставляет пространство для рисования, где можно напрямую управлять пикселями на экране, в то время как SVG основывается на векторной графике и использует XML для описания графических элементов.

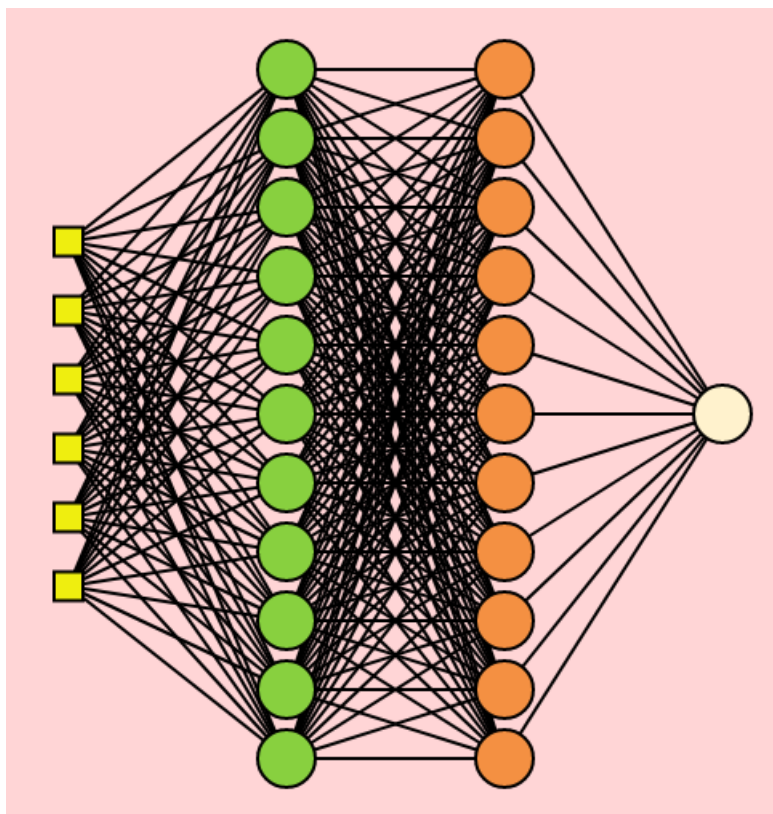


Рисунок 3 – Структура нейронной сети для прогнозирования распределения диоксида азота в атмосферном воздухе города Таганрога

## Заключение

При выборе технологии разработки для web важно выбрать актуальную технологию. Всего десяток лет назад подобные разработки массово делались на Flash с использованием ActionScript [8], с 2021 года поддержка Flash была прекращена.

JavaScript-библиотека `nndrawable.js` разработана в блокноте Notepad++ с помощью языка JavaScript. Ее легко можно встроить и использовать на любом сайте для изображения нейронных сетей, как в виде растровой, так и векторной графики.

Ее можно сочетать, например, с другими библиотеками, с помощью которых можно моделировать работу искусственной нейронной сети на языке программирования JavaScript [9].

При написании скрипта рисования нейронной сети используйте инструменты разработчика в браузере. Современные браузеры также предоставляют инструменты разработчика [10], которые можно использовать для выявления и исправления ошибок JavaScript. Эти инструменты предоставляют вам подробную информацию об ошибке и позволяют редактировать код и устранять проблемы непосредственно в приложении.

Выбор между SVG и Canvas зависит от конкретных требований проекта и опыта разработчика. Для создания простых и небольших графических элементов, SVG может быть более простым в использовании, так как он основывается на векторной графике, которая может быть легко масштабирована и изменена с помощью CSS и JavaScript.

В то же время, canvas может быть более простым для создания сложных анимаций, игр и других динамических графических элементов, которые могут быть более сложными для реализации в SVG.

## ЭЛЕКТРОННЫЙ НАУЧНЫЙ ЖУРНАЛ «ДНЕВНИК НАУКИ»

В целом, каждый из этих инструментов имеет свои преимущества и недостатки, и выбор зависит от конкретной задачи, которую нужно решить.

Исходный код выложен в репозитории на Github. GitHub – это веб-платформа, которая позволяет разработчикам хранить код и другие типы программных проектов, делиться ими и совместно работать над ними. Особенно удобен GitHub для публикации Интернет-проектов. Ознакомиться с демонстрацией возможностей библиотеки для рисования нейронных сетей можно по адресу <https://gevaraweb.github.io/nndrawable/>. Для ее использования достаточно любого современного браузера.

Приложение является бесплатным. Любой преподаватель может его использовать с учащимися на уроках или на внеурочных занятиях.

**Библиографический список:**

1. Вильданов А.Н., Шафеева Е.П. Построение объектов двумерной графики в HTML5 с помощью canvas // Информатика в школе, 2015, № 4(107). – С. 24-28. URL: [https://infojournal.ru/journals/school/school\\_04-2015/](https://infojournal.ru/journals/school/school_04-2015/).
2. Долгова Е.В., Файзрахманов Р.А., Курушин Д.С., Федоров А.Б., Хабибулин А.Ф., Шаронов А. А. Архитектура мобильного тренажера погрузочно-разгрузочного устройства // Инженерный вестник Дона, 2012, № 4 (часть1). URL: [ivdon.ru/magazine/archive/n4p1y2012/1327/](http://ivdon.ru/magazine/archive/n4p1y2012/1327/).
3. Гончаров А.Ю. Web-дизайн. HTML, JavaScript и CSS : карманный справочник. – Москва : КУДИЦ-Пресс, 2007, 319 с. – ISBN 5-91136-024-1.
4. Вильданов А.Н., Рекламно-техническое описание [Электронный ресурс]. 7с. URL: [http://ofernio.ru/rto\\_files\\_ofernio/25061.doc](http://ofernio.ru/rto_files_ofernio/25061.doc). – DOI 10.12731/ofernio.2022.25061.
5. McKenzie N. Is JavaScript an Object-Oriented Language? // Dr. Dobb's Journal: Software Tools for the Professional Programmer, 2001, Vol. 26, No 8. – P. 115-116.
6. Petersen P. Equivalence of approximation by convolutional neural networks and fully-connected networks / P. Petersen, F. Voigtlaender // Proceedings of the American Mathematical Society. – 2020. – Vol. 148. – No 4. – P. 1567-1581. – DOI 10.1090/proc/14789.
7. Плуготаренко Н. К., Варнавский А. Н. Применение нейронных сетей для построения модели прогнозирования состояния городской воздушной среды // Инженерный вестник Дона, 2012, №4-2. URL: [ivdon.ru/magazine/archive/n4p2y2012/1351/](http://ivdon.ru/magazine/archive/n4p2y2012/1351/).
8. Гарифуллин Р.И., Девяткин Е.М. Электронный комплекс виртуальных лабораторных установок по механике и молекулярной физике // Сборник Дневник науки | [www.dnevniknauki.ru](http://www.dnevniknauki.ru) | СМИ ЭЛ № ФС 77-68405 ISSN 2541-8327

## ЭЛЕКТРОННЫЙ НАУЧНЫЙ ЖУРНАЛ «ДНЕВНИК НАУКИ»

научных статей международной молодежной школы-семинара «Ломоносовские чтения на Алтае» : в 6 частях, Барнаул, 05–08 ноября 2013 года / Под редакцией Родионова Е.Д. Барнаул: Алтайский государственный университет, 2013. С. 309-311.

9. Баженов Р.И., Кочитов, М. Е. Использование библиотеки brainjs для разработки нейронной сети на языке программирования javascript // Международный журнал информационных технологий и энергоэффективности, 2019, Т. 4, № 4(14). С. 15-28.

10. Вильданов А.Н. 3D-моделирование на WebGL с помощью библиотеки Three.js : Учебное пособие. – Уфа : Башкирский государственный университет, 2014. – 113 с. – ISBN 9875747735606.

*Оригинальность 76%*