

## Искусственная нейронная сеть на JavaScript библиотеке BrainJS

*Кочитов Михаил Евгеньевич*

*Приамурский государственный университет им. Шолом-Алейхема*

*студент*

### Аннотация

В данной статье рассматривается создание искусственной нейронной сети, используя JavaScript библиотеку BrainJS. Также будет разработан веб-пример с интерактивной формой ввода значений для обучения и проверки искусственной нейронной сети.

**Ключевые слова:** искусственная нейронная сеть, BrainJS, JavaScript библиотека, обучение нейронной сети

## Artificial Neural Network at BrainJS JavaScript Library

*Kochitov Mikhail Evgenevich*

*Sholom-Aleichem Priamursky State University*

*student*

### Abstract

This article discusses the creation of an artificial neural network using the BrainJS JavaScript library. A web example will also be developed with an interactive form for entering values for training and testing an artificial neural network.

**Keywords:** artificial neural network, BrainJS, JavaScript library, neural network training

Искусственные нейронные сети очень востребованы по всему миру, так как от них технологии развиваются гораздо быстрее, применяя искусственный интеллект. По мере развития технологий появляются возможности создавать программы, которые будут делать те задачи, что делает сам человек. Искусственные нейронные сети в основном применяются в различных областях, например в распознавании различных фотографий и изображений, в подборке фильмов, которые интересны самому зрителю, в улучшении искусственного интеллекта и так далее.

В статье M. Fayed, M. Elhadary, H. A. Abderrahmane, B. N. Zakher рассматривается возможность прогнозирования частоты колебания гибкой нити по искусственной нейронной сети [1]. В статье M. O. Selbesoglu можно увидеть прогнозирование задержки тропосферной влаги по модели искусственной нейронной сети на основе метеорологических данных и данных GNSS [2]. R. A. Tenenbaum, F. O. Taminato, V. S. G. Melo в своей статье рассмотрели быструю аурализацию с использованием радиальных базисных функций типа методов искусственной нейронной сети [3]. В статье

W.Ye, X.Wang, Y.Liu рассматривается применение искусственной нейронной сети для прогнозирования динамических характеристик свободнопоршневого двигателя Стирлинга [4]. S. Delfani, M. Esmaeili, M. Karami в своей статье рассмотрели применение искусственной нейронной сети для прогнозирования производительности солнечного коллектора с прямым поглощением на основе наножидкости [5].

Целью статьи является создание искусственной нейронной сети, используя JavaScript библиотеку BrainJS. Также будет создан веб-пример с интерактивной формой ввода значений от пользователя для обучения и проверки нейронной сети.

При создании искусственной нейронной сети понадобится JavaScript библиотека BrainJS [6], позволяющая работать на языке программирования JavaScript с нейронными сетями различного рода в самом браузере. Также помимо библиотеки BrainJS понадобится еще JavaScript JQuery [7], упрощающий немного написание JavaScript кода. Для разработки нейронной сети не потребуется локальный сервер, вся работа будет осуществляться только на клиенте самого браузера.

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Нейронная сеть BrainJS</title>
    <script src="jquery.min.js" type="text/javascript"></script>
    <script src="brain-browser.min.js" type="text/javascript"></script>
  </head>
  <body>
  </body>
</html>
```

Рисунок 1 – HTML код шапки загрузки библиотеки BrainJS и скрипта JQuery

На рисунке 1 изображен HTML код шапки веб-страницы, в котором видна загрузка двух JS скриптов - это сама библиотека по нейронным сетям BrainJS (на рисунке файл называется «brain-browser.min.js») и библиотека JQuery (на рисунке файл называется «jquery.min.js»).

Далее необходимо написать HTML код содержимого самой страницы, чтобы визуально в ней отображались некоторые элементы и сама интерактивная форма.

```

<div>
  <p>Обучение нейронной сети</p>
  <table>
    <tr>
      <th>Вход 1</th>
      <th>Вход 2</th>
      <th>Вход 3</th>
      <th>Выход</th>
    </tr>
    <tr>
      <td><input type="number" id="in_1_1" min="0" max="1"></td>
      <td><input type="number" id="in_1_2" min="0" max="1"></td>
      <td><input type="number" id="in_1_3" min="0" max="1"></td>
      <td><input type="number" id="out_1" min="0" max="1"></td>
    </tr>
    <tr>
      <td><input type="number" id="in_2_1" min="0" max="1"></td>
      <td><input type="number" id="in_2_2" min="0" max="1"></td>
      <td><input type="number" id="in_2_3" min="0" max="1"></td>
      <td><input type="number" id="out_2" min="0" max="1"></td>
    </tr>
    <tr>
      <td><input type="number" id="in_3_1" min="0" max="1"></td>
      <td><input type="number" id="in_3_2" min="0" max="1"></td>
      <td><input type="number" id="in_3_3" min="0" max="1"></td>
      <td><input type="number" id="out_3" min="0" max="1"></td>
    </tr>
    <tr>
      <td><input type="number" id="in_4_1" min="0" max="1"></td>
      <td><input type="number" id="in_4_2" min="0" max="1"></td>
      <td><input type="number" id="in_4_3" min="0" max="1"></td>
      <td><input type="number" id="out_4" min="0" max="1"></td>
    </tr>
    <tr>
      <td><input type="number" id="in_5_1" min="0" max="1"></td>
      <td><input type="number" id="in_5_2" min="0" max="1"></td>
      <td><input type="number" id="in_5_3" min="0" max="1"></td>
      <td><input type="number" id="out_5" min="0" max="1"></td>
    </tr>
  </table>
  <button onclick="train_neuron_net()">Обучить нейронную сеть</button>
  <div id="result_error"></div>
</div>

```

Рисунок 2 – HTML код интерактивной формы для обучения нейронной сети

```

<div>
  <p>Проверка нейронной сети</p>
  <table>
    <tr>
      <th>Вход 1</th>
      <th>Вход 2</th>
      <th>Вход 3</th>
      <th>Выход</th>
    </tr>
    <tr>
      <td><input type="number" id="in_1" min="0" max="1"></td>
      <td><input type="number" id="in_2" min="0" max="1"></td>
      <td><input type="number" id="in_3" min="0" max="1"></td>
      <td><input type="number" id="out" readonly min="0" max="1"></td>
    </tr>
  </table>
  <button onclick="check_neuron_nnet()">Проверить нейронную сеть</button>
</div>

```

Рисунок 3 – HTML код интерактивной формы для проверки нейронной сети

На рисунках 2 и 3 представлены HTML коды интерактивных форм обучения и проверки нейронной сети. Сама форма представлена в виде

таблицы, где в каждой ячейке имеется поле для ввода числового значения в диапазоне от 0 до 1. Таблица состоит из 4-ех строк и 4-ех столбцов, это значит, что для обучения нейронной сети в примере будет использоваться 4 набора данных, а столбцы означают три входных поля, а четвертое – выходное. Также под двумя таблицами добавлены элементы кнопок, при клике которых они вызывают соответствующие функции, которые будут рассмотрены ниже

```
var main_nnet;
function train_neuron_net() {
    var hidden_layers = [2];
    hidden_layers[0] = 3;
    hidden_layers[1] = 3;
    input_data = new Array(5);
    output_data = new Array(5);
    train_data = new Array(5);
    for (var i = 0; i < 5; i++) {
        input_data[i] = new Array(3);
        train_data[i] = {};
    }
    for (var i = 0; i < 5; i++) {
        for (var j = 0; j < 3; j++) {
            input_data[i][j] = $('#in_'+(i+1)+'_'+(j+1)).val();
        }
        output_data[i] = $('#out_'+(i+1)).val();
        train_data[i]['input'] = input_data[i];
        train_data[i]['output'] = output_data[i];
    }
    const config = {
        hiddenLayers: hidden_layers,
        activation: 'sigmoid',
    };
    const net = new brain.NeuralNetwork(config);
    err = net.train(train_data);
    main_nnet = net;
    $('#result_error').text("Нейронная сеть обучена! Среднеквадратичная ошибка: "+err.error);
}
```

Рисунок 4 – JavaScript код функции «train\_neuron\_net()»

На рисунке 4 изображен код, написанный на языке программирования JavaScript в котором расписано содержимое функции «train\_neuron\_net()». Данная функция дает возможность обучать полностью искусственную нейронную сеть над данными, которые были введены пользователем в таблице формы для обучения нейронной сети (рис. 2). Если рассматривать более подробно код, то вне функции объявлена глобальная переменная main\_nnet, которая будет являться объектом нейронной сети, чтобы к обученной нейронной сети обратиться с другой функции. Массив input\_data используется для заполнения в него входных значений, а массив output\_data – для выходных значений, массив train\_data используется исключительно для формирования правильной заготовки набора значений перед внедрением ее в нейронную сеть для обучения. Также для нейронной сети разработан конфиг, в котором указывается массив скрытых слоев (hidden\_layers) и функция активации sigmoid. Для создания объекта нейронной сети из библиотеки BrainJS используется метод «brain.NeuralNetwork», а функция «net.train» позволяет обучить нейронную сеть над данными, которые были сформированы в массив train\_data.

```
function check_neuron_nnet() {
  enter_input = [{}];
  for (var i = 0; i < 3; i++) {
    enter_input[i] = $('#in_'+(i+1)).val();
  }
  const result_nnet = main_nnet.run(enter_input);
  $('#out').val(result_nnet[0].toFixed(3));
}
```

Рисунок 5 – JavaScript код функции «check\_neuron\_nnet()»

На рисунке 5 представлен маленький JavaScript код, в котором написано содержимое функции «check\_neuron\_nnet()», в которой работа заключается в проверки обученной нейронной сети. При рассматривании подробно кода в начале функции объявляется массив enter\_input, который берет входные значения, введенные пользователем на форме для проверки нейронной сети (рис. 3). Далее с помощью функции «main\_nnet.run» запускается проверка обученной нейронной сети и выдается результат, который приблизительно должен быть правильным.

Для обучения искусственной нейронной сети используем данные из задачи о зонтике, где нужно выяснить при какой погоде можно взять зонтик [8]. Теперь опираясь на данную задачу, составим таблицу всех исходов

Таблица 1 – Набор данных задачи о зонте

Уже идет дождь?	Влажно?	Облачно?	Взять зонтик?
Да	Нет	Нет	Да
Нет	Да	Да	Да
Нет	Да	Нет	Нет
Нет	Нет	Да	Нет
Да	Да	Да	Да

Сформировав таблицу набора данных, можно перейти к обзору готового интерфейса веб-страницы нейронной сети с двумя интерактивными табличными формами и ввести в таблицу данные из таблицы 1

## Нейронная сеть на JS библиотеке BrainJS

Обучение нейронной сети

Вход 1	Вход 2	Вход 3	Выход
1	0	0	1
0	1	1	1
0	1	0	0
0	0	1	0
1	1	1	1

Обучить нейронную сеть

Нейронная сеть обучена! Среднеквадратичная ошибка: 0.004987674296010003

Проверка нейронной сети

Вход 1	Вход 2	Вход 3	Выход
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Проверить нейронную сеть

Рисунок 6 – Готовый веб-пример искусственной нейронной сети с добавленным набором данных задачи

На рисунке 6 изображен веб-пример искусственной нейронной сети в браузере. На этой веб-странице расположились две табличные интерактивные формы, две кнопки и несколько заголовков. В первой табличной форме были введены значения из данной задачи, соответствующие из таблицы 1. Также при нажатии кнопки «Обучить нейронную сеть» появилось уведомление о том, что нейронная сеть обучена и показана среднеквадратичная ошибка, предусматривающая, что нейронная сеть хорошо обучилась, так как ее ошибка приблизительно равна 0.004.

Далее осталось проверить обученную нейронную сеть над каждым набором данных и после составить итоговую таблицу для сверки результатов.

На рисунке 8 изображен результат проверки нейронной сети на пяти наборах данных из задачи о зонтике. Опираясь на рисунок 8 и на данные из таблицы 1, составим таблицу результатов, чтобы увидеть наглядно результат

Таблица 2 – Результаты обученной искусственной нейронной сети

Вход 1 (Уже идет дождь?)	Вход 2 (Влажно?)	Вход 3 (Облачно?)	Выход (Взять зонтик?)	Исходный результат
1	0	0	0.947	1
0	1	1	0.917	1
0	1	0	0.086	0
0	0	1	0.086	0
1	1	1	0.988	1

## Проверка нейронной сети

Вход 1	Вход 2	Вход 3	Выход
<input type="text" value="1"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	0,947
<input type="button" value="Проверить нейронную сеть"/>			
Вход 1	Вход 2	Вход 3	Выход
<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	0,917
<input type="button" value="Проверить нейронную сеть"/>			
Вход 1	Вход 2	Вход 3	Выход
<input type="text" value="0"/>	<input type="text" value="1"/>	<input type="text" value="0"/>	0,086
<input type="button" value="Проверить нейронную сеть"/>			
Вход 1	Вход 2	Вход 3	Выход
<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="1"/>	0,086
<input type="button" value="Проверить нейронную сеть"/>			
Вход 1	Вход 2	Вход 3	Выход
<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	0,988
<input type="button" value="Проверить нейронную сеть"/>			

Рисунок 8 – Проверка обученной искусственной нейронной сети

На рисунке 8 представлена таблица результатов, которые выдала искусственная нейронная сеть, обученная над набором данных из задачи о зонтике. В таблице на вход подавались значения 0 и 1, которые означают логические значения: 0 – Нет (ложь), 1 – Да (истина). Также сверяя два последних столбца (Выход и исходный результат) можно убедиться, что значения приблизительно равны и это значит, что нейронная сеть выдала правильные результаты.

Таким образом, была создана искусственная нейронная сеть, используя JavaScript библиотеку BrainJS. Также был разработан веб-пример нейронной сети, в которой возможно обучать и проверять нейронную сеть над пятью наборами данных, в нашем случае взяты данные из задачи о зонтике. Можно предположить, что данный веб-пример можно улучшить визуальным образом так, что можно добавлять больше строк и столбцов в таблице для добавления еще большего количества набора данных и входных значений, позволяющих более глубоко обучить искусственную нейронную сеть.

**Библиографический список**

1. Fayed M., Elhadary M., Abderrahmane H. A., Zakher B. N. The ability of forecasting flapping frequency of flexible filament by artificial neural network // Alexandria Engineering Journal, T. 58, 2019, C. 1367-1374
2. Selbesoglu M. O. Prediction of tropospheric wet delay by an artificial neural network model based on meteorological and GNSS data // Engineering Science

- and Technology, an International Journal, In press, corrected proof, 2019
3. Tenenbaum R. A., Taminato F. O., Melo V. S. G. Fast auralization using radial basis functions type of artificial neural network techniques // Applied Acoustics, T. 157, 2020, С. 106993
  4. Ye W., Wang X., Liu Y. Application of artificial neural network for predicting the dynamic performance of a free piston Stirling engine // Energy, T. 194, 2020, С. 116912
  5. Delfani S., Esmaeili M., Karami M. Application of artificial neural network for performance prediction of a nanofluid-based direct absorption solar collector // Sustainable Energy Technologies and Assessments, T. 36, 2019, С. 100559
  6. GitHub – BrainJS Neural networks in JavaScript // URL: <https://github.com/BrainJS/brain.js> (дата обращения 14.01.2020)
  7. JQuery Write less, do more // URL: <https://jquery.com/> (дата обращения 14.01.2020)
  8. Нейронная сеть для задачи о зонтике // URL: [https://ozlib.com/868579/informatika/neuronnaya\\_set\\_zadachi\\_zontike](https://ozlib.com/868579/informatika/neuronnaya_set_zadachi_zontike) (дата обращения 14.01.2020)
  9. Искусственные нейронные сети простыми словами – Хабр // URL: <https://habr.com/ru/post/369349/> (дата обращения 14.01.2020)
  10. Пишем свою нейросеть: пошаговое руководство // URL: <https://proglib.io/p/neural-nets-guide> (дата обращения 14.01.2020)