УДК 004

Использование JavaScript для создания нейронной сети

Шайдуров Александр Александрович
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной работе проведено исследование возможности использования нейронной сети. Данная нейронная сеть была создана на языке JavaScript. Также в работе описаны исследования на тему нейронных сетей.
Ключевые слова: Нейронная сеть, JavaScript.

Using JavaScript to create a neural network

Shaidurov Aleksandr Aleksandrovich
Sholom-Aleichem Priamursky State University
Student

Abstract
In this paper, we study the possibility of using a neural network. This neural network was created in JavaScript. The paper also describes research on the topic of neural networks.
Keywords: Neural network, JavaScript.

Современные технологии позволяют решать множество задач. Одним из способов решения задач, является использование нейронных сетей. Главная особенность нейросетей, они обладают возможностью обучения.

Целью исследования является показать работу нейронной сети, созданной с помощью языка программирования JavaScript.

В данный момент один из самых языков, является JavaScript. JavaScript — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили.

В работе В.Г. Варданян описываются разработанные методы оптимизации динамических многоуровневых компиляторов с учетом информации о профиле выполнения программы. Метод был реализован в динамическом компиляторе языка JavaScript, разработанном компанией Google. Использование профиля выполнения программы позволяет оптимизировать программу для конкретных входных данных [1]. В статье Г.И. Додонова и И.И. Холкина представлено описание возможностей матричных преобразований видеопотоков на основе применения платформы HTML5 и JavaScript без использования дополнительных программ и библиотек. Даны краткие характеристики используемых инструментов, описаны преимущества такого подхода, рассматриваются примеры создания

видеоэффектов различной сложности [2]. В статье Н.А. Бруевича описана реализация инструмента классификации продуктов питания, использующего интеллектуальный анализ данных. Основной задачей является обучение алгоритма на основе входных значений продуктов питания методами машинного обучения. Важным этапом является проверка признаков продуктов питания на корреляцию с целью уменьшения значимых признаков, что ускорит выполнение обучения программы. Для проверки точности обучения модели использовался метод перекрестной проверки точности классификации [3]. В статье В.Н. Белова, А.И. Ковалёвы и С.А. Новикова рассмотрен один из самых популярных и часто используемых Java скриптов – Slider. Говоря другими словами – это обычный ползунок, используемый для выставления нужных значений или общего уровня каких-либо значений. Данный плагин получил огромное распространение в сети, благодаря своей относительной простоте и большому функционалу. Плагин Slider позволяет превращать блочные элементы, такие как div`ы, в элементы управления, которые называются ползунками. Установить нужный диапазон значений, изменение громкости, ценовые рамки, все это чаще всего реализовывается с помощью Slider`a. Передвигать кнопки ползунков можно с помощью мыши, или, если ползунок находится в фокусе, с помощью стрелок на клавиатуре. Будут рассмотрены все особенности внедрения и инициализации плагина, основные положительные стороны и существующие на данный момент минусы. Для использования данного материала требуется минимальное знание JavaScript и HTML [4]. В работе В.В. Плющева, В.В. Бронской и Г.В. Мануйко описана разработка приложения для прогнозирования модуля релаксации напряжений блок-сополимера пропилена и этилена с помощью искусственных нейронных сетей. Составлены обучающие и тестовые выборки сети по экспериментальным данным. Искусственная нейронная сеть была реализована на языке Python 3 с использованием библиотеками Keras в интерактивной оболочке Jupyter Notebook. Также были использованы другие библиотеки, такие как NumPy, Matplotlib и Scikit-Learn [5]. П.А. Сахнюк в своей работе рассмотрел возможности, предоставляемые облачным сервисом Google Colab для изучения технологий машинного обучения и нейронных сетей в образовательных организациях [6].

В данной работе описана нейронная сеть, работающая с помощью библиотеки «brain.js».

Создание холста, на котором будет происходить действие. Данный холст будет пикселизирован, то есть разделён на несколько частей. Размер холста будет высотой 500 и шириной 500 пикселей. Переменная «is_mouse_down» создана, для возможности создания рисунка на холсте.

```
<script>
    function DCanvas(el)
    {
        const ctx = el.getContext('2d');
        const pixel = 20;

        let is_mouse_down = false;

        canv.width = 500;
        canv.height = 500;
```

Рис.1. Создания холста

Добавлены три функции. Первая для того, чтобы не повторять каждый раз одни те же действия, которые требуются, чтобы рисовать линии и ячейки в сетке. Вторая для отрисовки ячеек в сетке. Третья для отчистки сетки. Методы drawLine и drawCell необходимы для того, чтобы не повторять каждый раз одни и те же действия требующие, чтобы рисовать линию и отрисовки ячеек в сетке. Метод «clear» при вызове будет отчищать сетку.

```
this.drawLine = function(x1, y1, x2, y2, color = 'gray') {
    ctx.beginPath();
    ctx.strokeStyle = color;
    ctx.lineJoin = 'miter';
    ctx.lineWidth = 1;
    ctx.moveTo(x1, y1);
    ctx.lineTo(x2, y2);
    ctx.stroke();
}

this.drawCell = function(x, y, w, h) {
    ctx.fillStyle = 'blue';
    ctx.strokeStyle = 'blue';
    ctx.lineJoin = 'miter';
    ctx.lineWidth = 1;
    ctx.rect(x, y, w, h);
    ctx.fill();
}

this.clear = function() {
    ctx.clearRect(0, 0, canv.width, canv.height);
}
```

Рис.2. Создание методов

Для создания сетки с помощью, которой будет проходить обучение и работа нейросети, понадобится следующий код. В ней элементам, которые будут попадать и не попадать в клетки, будут присвоены 0 и 1, нейросеть будет анализировать только 0 и 1.

```
this.drawGrid = function() {
    const w = canv.width;
    const h = canv.height;
    const p = w / pixel;

    const xStep = w / p;
    const yStep = h / p;

    for( let x = 0; x < w; x += xStep )
    {
        this.drawLine(x, 0, x, h);
    }

    for( let y = 0; y < h; y += yStep )
    {
        this.drawLine(0, y, w, y);
    }
}
```

Рис.3. Создание сетки

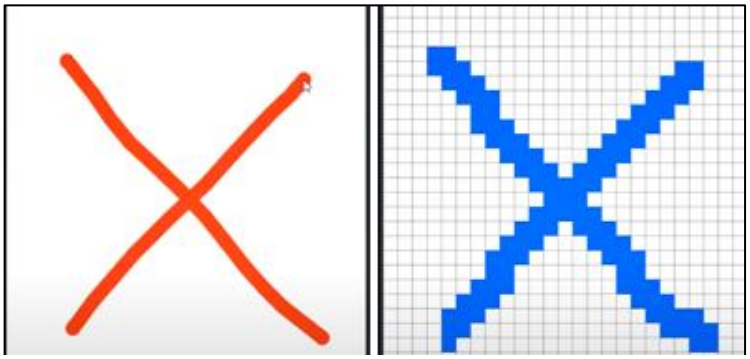

Рис.4. Обработка изображения сеткой

Для создания отрисовки, при попытки создании изображения, используется следующий код.

```
el.addEventListener('mousedown', function(e) {
    is_mouse_down = true;
    ctx.beginPath();
})

el.addEventListener('mouseup', function(e) {
    is_mouse_down = false;
})

el.addEventListener('mousemove', function(e) {
    if( is_mouse_down )
    {
        ctx.fillStyle = 'red';
        ctx.strokeStyle = 'red';
        ctx.lineWidth = pixel;

        ctx.lineTo(e.offsetX, e.offsetY);
        ctx.stroke();

        ctx.beginPath();
        ctx.arc(e.offsetX, e.offsetY, pixel / 2, 0, Math.PI * 2);
        ctx.fill();

        ctx.beginPath();
        ctx.moveTo(e.offsetX, e.offsetY);
```

Рис.5. Код для отрисовки изображения

Для того чтобы нейросеть обучалась, она должна понимать, что изображено на холсте. Для этого она будет считывать ячейки, которых касается изображение и присваиваться им единицу, а тем, которых не коснулось изображение нуль.

```
this.calculate = function(draw = false) {
    const w = canv.width;
    const h = canv.height;
    const p = w / pixel;

    const xStep = w / p;
    const yStep = h / p;

    const vector = [];
    let __draw = [];

    for( let x = 0; x < w; x += xStep )
    {
        for( let y = 0; y < h; y += yStep )
        {
            const data = ctx.getImageData(x, y, xStep, yStep);

            let nonEmptyPixelsCount = 0;
            for( i = 0; i < data.data.length; i += 10 )
            {
                const isEmpty = data.data[i] === 0;

                if( !isEmpty )
                {
                    nonEmptyPixelsCount += 1;
                }
            }

            if( nonEmptyPixelsCount > 1 && draw )
            {
                __draw.push([x, y, xStep, yStep]);
            }

            vector.push(nonEmptyPixelsCount > 1 ? 1 : 0);
        }
    }
```

Рис.6. Код обучения нейросети

Константы необходимы для определения шага по двум осям. Массив «vextor» необходим для хранения результата подсчёта, в нём будут храниться нули и единицы при переводе изображения. Массив «__draw», нужен для возможности создавать рисунок после подсчёта.

```
vector.push(nonEmptyPixelsCount > 1 ? 1 : 0);
```

Рис.7. Присвоение пикселям сетки нуля и единицы

В строке присваивается условие, если в клетки больше заполненных пикселей, присваивается «1», если незаполненных, то «0».

```
if( draw )
{
    this.clear();
    this.drawGrid();

    for( _d in __draw )
    {
        this.drawCell( __draw[_d][0], __draw[_d][1], __draw[_d][2], __draw[_d][3] );
    }
}
```

Рис.8. Отрисовка

Передача аргументов «__draw» для создания отрисовки.

Для управления задействованы клавиши «c» - отчистка, «v» - внесение рисунка нейросеть, «b» - проверка изображения нейросетью.

```
const d = new DCanvas(document.getElementById('canv'));

document.addEventListener('keypress', function(e) {
    if( e.key.toLowerCase() == 'c' )
    {
        d.clear();
    }

    if( e.key.toLowerCase() == 'v' )
    {
        vector = d.calculate(true);

        //train
        if( confirm('Positive?') )
        {
            train_data.push({
                input: vector,
                output: {positive: 1}
            });
        } else
        {
            train_data.push({
                input: vector,
                output: {negative: 1}
            });
        }
    }

    if( e.key.toLowerCase() == 'b' )
    {
        net = new brain.NeuralNetwork();
        net.train(train_data, {log: true});

        const result = brain.likely(d.calculate(), net);
        alert(result);
    }
```

Рис.9. Управление

Для того, что сеть могла отличать изображения, после создания изображения и нажатия клавиши «v», сетью будет открыто окно с вопросом «Positive?». С помощью строк «output: {positive: 1}» и «output: {negative: 1}» и будет присвоено, что рисунок является необходимым или нет.

Метод «brain.likely» позволит определить какой объект нарисован. Первым аргументом передаётся текущий вектор, который необходимо проверить, вторым сама нейронная сеть.

```
if( e.key.toLowerCase() == 'b' )
{
    net = new brain.NeuralNetwork();
    net.train(train_data, {log: true});

    const result = brain.likely(d.calculate(), net);
    alert(result);
}
```

Рис.10. Вывод результата

Если обучая сеть нарисовать несколько одинаковых рисунков и подтвердить «ok», а также несколько других одинаковых изображения нажав «отмена». То при следующем создании изображения и нажатия клавиши «b», сеть выдаст результат.

Рис.11. Окно при обучении нейросети

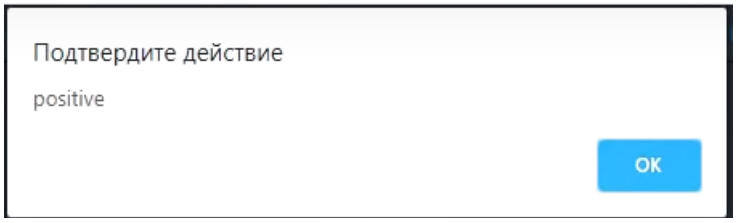

Рис.12. Положительный ответ нейросети

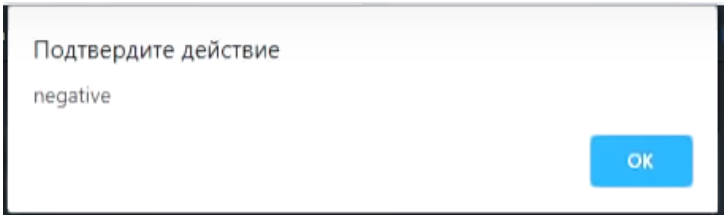

Рис.13. Негативный ответ нейросети

Таким образом была описана нейронная сеть, написанная на языке JavaScript. Данная нейронная сеть способна различать два различных типа изображения. Описана работа и функционал данной нейронной сети. Также в данной работе есть анализ работ, где описаны создание и работа других нейронных сетей.

Библиографический список

1. Варданян В.Г. Методы оптимизации программ на языке JavaScript, основанные на статистике выполнения программы // Труды института системного программирования РАН. №1. 2016. С. 5-10.
2. Додонов Г.И., Холкин И.И. Матричные преобразования видеопотоков с использованием платформы HTML5 И JavaScript // Научный альманах №3. 2016. С.71-74.
3. Бруевич Н.А. Реализация классификатора продуктов питания с помощью метода машинного обучения // Научное обозрение. Педагогические науки, № 4-3. 2019. С.30-34.
4. Белов В.Н., Ковалёв А.И., Новиков С.А. Современные наукоемкие технологии // Современные наукоемкие технологии. №2. 2016. С.224-228.
5. Плющев В.В., Бронская В.В., Мануйко Г.В. Нейросетевая модель прогнозирования модуля релаксации напряжений образцов блок-сополимера пропилена и этилена // XXIV Туполевские чтения (Школа молодых ученых). 2019. С. 265-268.
6. Сахнюк П.А. Возможности Google Colab для изучения технологий машинного обучения и нейронных сетей // Информатизация непрерывного образования – 2018, 2018. С.586-588.