

УДК 004.438:004.85

## АКТУАЛЬНОСТЬ ИСПОЛЬЗОВАНИЯ ЯЗЫКА JAVASCRIPT ДЛЯ РАЗРАБОТКИ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ



**Г.А. Пискун**

Доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент  
[piskunbsuir@gmail.com](mailto:piskunbsuir@gmail.com)



**В.Ф. Алексеев**

Доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент  
[alexvikt.minsk@gmail.com](mailto:alexvikt.minsk@gmail.com)



**Т.М. Воронко**

Инженер-программист Центра информатизации и инновационных разработок БГУИР, магистрант БГУИР  
[voronko232001@gmail.com](mailto:voronko232001@gmail.com)

### **Г.А. Пискун**

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с моделированием и оптимальным проектированием информационно-компьютерных систем, организацией учебного и научно-исследовательского процессов в техническом университете.

### **В.Ф. Алексеев**

Окончил Минский радиотехнический институт. Область научных интересов связана с разработкой методов и алгоритмов построения информационно-компьютерных систем, организацией учебного и научно-исследовательского процессов в техническом университете.

### **Т.М. Воронко**

Магистрант 1 курса специальности «Электронные системы и технологии» кафедры проектирования информационно-компьютерных систем. Инженер-программист в отделе инновационных разработок в сфере образования Центра информатизации и инновационных разработок Белорусского государственного университета информатики и радиоэлектроники.

**Аннотация.** Выполнен анализ применимости и актуальности использования языка программирования *JavaScript* для разработки моделей машинного обучения. Рассмотрены основные преимущества и недостатки данного метода работы с искусственным интеллектом в сравнении с аналогами, такими как *Python*, а также описаны особенности *JavaScript*-библиотеки *TensorFlow.js*, с предоставлением примера программы для построения модели, предсказывающей координаты точки по заданным параметрам. Установлено, что использование *JavaScript* может быть полезно для легких и нетребовательных к ресурсам кроссплатформенных веб-приложений, использующих модели машинного обучения.

**Ключевые слова:** машинное обучение, библиотека, искусственный интеллект.

### **Введение.**

На сегодняшний день, большинство разработчиков для создания моделей машинного обучения отдадут предпочтение таким языкам, как *Python*, *R* и *C++*, в виду зрелости их экосистем по работе с искусственным интеллектом (ИИ). Однако последние разработки в области *JavaScript* делают этот язык все более привлекательным для разворачивания моделей машинного обучения. Вполне возможно, что через несколько лет мы увидим значительный рост популярности использования данного языка для разработки подобных моделей, особенно по мере того, как ноутбуки и мобильные устройства становятся все более мощными, а популярность самого *JavaScript* растет [1].

Так как *JavaScript* – это в первую очередь язык программирования, который даёт возможность реализовывать сложное поведение веб-страницы, как, к примеру, динамически изменяющийся *UI* или управление бизнес-логикой, внедрение машинного обучения в проекты,

написанные на данном языке, может быть очень полезно, положительно влияя на скорость и простоту разработки, а также на дальнейшее управление этими проектами [2].

Рассмотрим основные недостатки и преимущества использования данного языка для разработки моделей машинного обучения.

### **Недостатки и преимущества использования *JavaScript*.**

В качестве основных недостатков использования *JavaScript* для машинного обучения, препятствующих увеличению его популярности в данной сфере, на сегодняшний день можно обозначить следующие характеристики:

1. **Низкая скорость работы с данными.** *JavaScript* – это однопоточный язык, а это означает, что модель машинного обучения может получать только одну часть информации за раз, что делает процесс обучения для этих моделей более длительным по сравнению с аналогами, написанными на других языках [3].

Однопоточность также может вызывать задержки или зависания системы при возникновении трудностей с выполнением определенного кода. Это одна из основных причин, по которой многие разработчики выбирают *Python*, который может похвастаться ускоренным процессом обучения и, как правило, работает быстрее, что ускоряет разработку приложений [3].

На рисунке 1 представлен график, отображающий время, которое тратят модели, написанные на *JavaScript* и *Python*, на обучение и прогнозирование [4], из которого можно сделать вывод, что модели на *Python* работают в разы быстрее, чем на *JavaScript*.

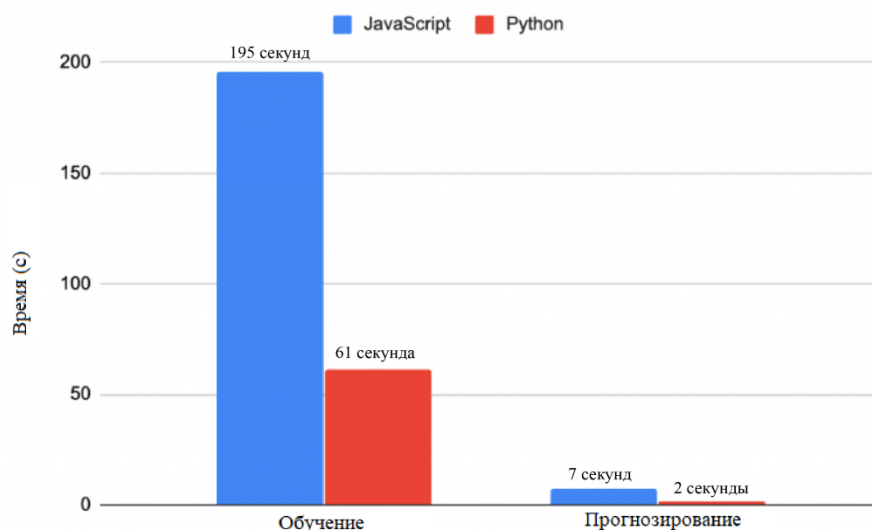


Рисунок 1. Время, затраченное на обучение и прогнозирование моделями на *JavaScript* и *Python*

2. **Малая коллекция библиотек.** Несмотря на то, что в *JavaScript* быстро растет число библиотек для машинного обучения, многие языки обладают гораздо более обширной коллекцией встроенных библиотек, которые прошли проверку временем и хорошо протестированы [3].

3. **Работа только с малым объемом данных.** Еще одним недостатком *JavaScript* является то, что существуют ограничения на объем данных, к которым он может обращаться, хранить и использовать. *JavaScript*-код не имеет доступа по умолчанию к файловой системе основного браузера, что также может привести к ограничениям в размерах файлов [3].

Говоря о преимуществах использования данного языка программирования в машинном обучении, необходимо выделить следующие показатели:

**1. Популярность и доступность.** *JavaScript* является одним из самых популярных и широко используемых языков программирования в мире, что повышает его доступность в сфере разработки приложений для машинного обучения. В то время как большинство других языков программирования по умолчанию не поддерживаются рядом устройств, почти каждый современный телефон и ноутбук поддерживает *JavaScript* без необходимости установки дополнительного программного обеспечения. Из-за его широкой популярности сообщество поддержки *JavaScript* также очень обширно и активно [3].

Поскольку многие люди уже используют *JavaScript*, и так как его легко освоить даже тем, у кого нет опыта, он может быть менее пугающим для новичков в машинном обучении, чем другие, менее знакомые языки. Машинное обучение для *JavaScript* также очень быстро и легко освоить, если вы уже являетесь разработчиком *JavaScript*. В конечном счете, команде разработчиков, уже использующей *JavaScript*, гораздо проще изучить алгоритмы машинного обучения на этом языке, чем изучать совершенно другой, что в долгосрочной перспективе может сэкономить время и ресурсы [3].

**2. Приватность данных.** Кибербезопасность является одной из главных забот предприятий и разработчиков программного обеспечения, поэтому компании проводят оценки безопасности разработки, чтобы снизить угрозу утечки данных, связанной с кодом. Как правило, куда более безопасно запускать машинное обучение на устройстве пользователя, а не на сервере [3].

Поскольку большинство приложений машинного обучения основаны на архитектуре клиент-сервер и передают информацию обратно на сервер, личные сообщения, изображения и электронные письма могут попасть в нежелательные руки на сервере, на котором работает модель машинного обучения. *JavaScript*, с другой стороны, не зависит от сервера для возможностей машинного обучения и может работать без отправки вашей информации третьей стороне [3].

**3. Кастомизация машинного обучения.** *JavaScript* также упрощает настройку моделей машинного обучения для отдельных пользователей по сравнению с приложениями, разработанными на других языках, что делает его идеальным для проектов, предполагающих широкое развертывание и множество разных пользователей. Доказано, что персонализированная целевая реклама способствует развитию бизнеса за счет повышения лояльности клиентов, увеличения продаж и предоставления вам данных о ваших клиентах, которые помогут вам узнать о вашем рынке и о том, что они ищут [3].

Для серверного приложения машинного обучения настройка будет означать хранение на сервере разных моделей машинного обучения для каждого пользователя, занимающих много места. С помощью *JavaScript* у вас есть возможность разместить копию модели машинного обучения на устройстве каждого пользователя. Отсюда вы можете использовать обучающие библиотеки для настройки модели на каждом уникальном устройстве. Это делает приложение более доступным для пользователей и снижает вероятность того, что вам придется покупать несколько дорогих серверов [3].

На рисунке 2 представлена схема распределения моделей на устройствах конечных пользователей.



Рисунок 2. Схема распределения моделей на устройствах конечных пользователей

**4. Простая интеграция машинного обучения в веб- и мобильных приложениях.** Для JavaScript уже существует богатый набор кроссплатформенных инструментов разработки мобильных приложений, таких как *Cordova* и *Ionic* [4].

Эти инструменты стали очень популярными, поскольку позволяют один раз написать код и развернуть его для устройств *iOS* и *Android*. Чтобы сделать код совместимым с различными операционными системами, средства кроссплатформенной разработки запускают «веб-представление» — объект браузера, который может запускать код *JavaScript* и может быть встроен в собственное приложение целевой операционной системы. Эти объекты браузера поддерживают библиотеки машинного обучения *JavaScript* [4].

Единственным исключением является *React Native*, популярная кроссплатформенная среда разработки мобильных приложений, которая не использует веб-просмотр для запуска приложений. Однако, учитывая популярность мобильных приложений машинного обучения, *Google* выпустила специальную версию *TensorFlow.js* для *React Native* [4].

В готовое мобильное приложение можно добавить встроенный объект браузера (например, *WKWebView* в *iOS*) для интеграции с кодом машинного обучения, а готовую версию приложения машинного обучения для браузера, можно легко перенести в мобильное приложение с небольшими изменениями или без них [4].

На сегодняшний день самым распространенным инструментом для разработки моделей машинного обучения на *JavaScript* является библиотека *TensorFlow.js*.

#### **Библиотека для машинного обучения на *JavaScript* – *TensorFlow.js***

Данная библиотека с открытым исходным кодом, разработанная компанией *Google*, предназначена для разворачивания машинного обучения в среде браузера и является основным проводником в мир ИИ для *JavaScript*-разработчика. Она максимально использует вышеперечисленные достоинства языка *JavaScript* в сфере машинного обучения, что открывает уникальные и ранее недоступные возможности для разработчиков.

Библиотека позволяет создавать новые модели машинного обучения с нуля, также как управлять и модифицировать уже существующие. Модель, созданная с помощью *TensorFlow* на *Python*, может быть конвертирована для работы в браузере [5].

На рисунке 3 представлены примеры проектов с официального сайта библиотеки, созданных с помощью нее [6].

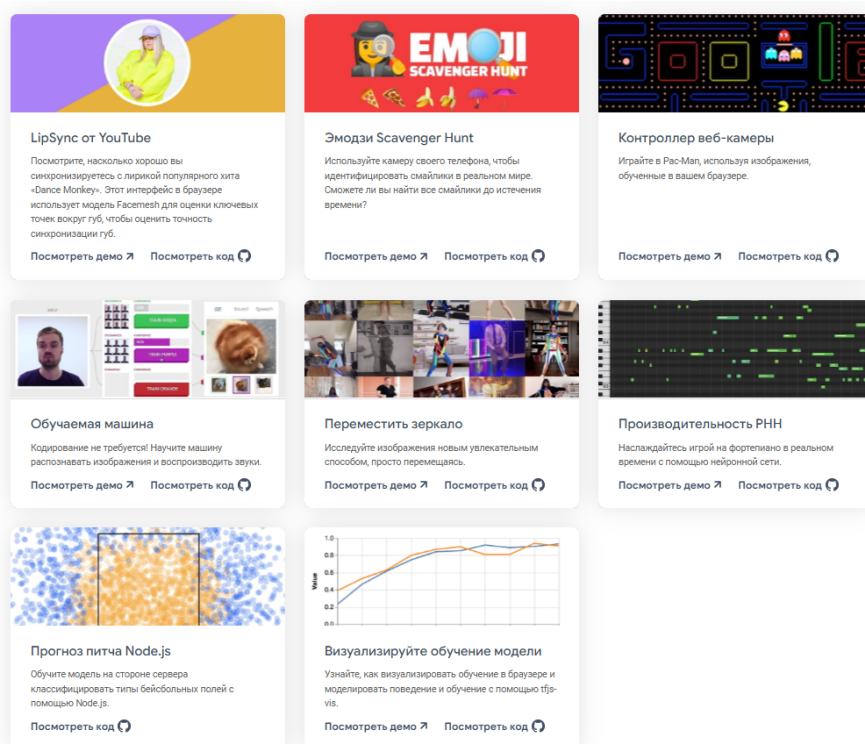


Рисунок 3. Примеры проектов, созданных с использованием библиотеки *TensorFlow.js*

Для обучения модели обычно как разработчику, так и конечному потребителю моделей машинного обучения, требуется установка соответствующих библиотек. Однако в *TensorFlow.js* интерфейсом является браузер, поэтому при подключении скриптового файла исчезает необходимость в установке библиотек и отслеживании зависимостей, все «доставляется на дом» [7].

Строение библиотеки можно представить в виде оболочечной структуры – ядерного *API* и покрывающих его высокоуровневых слоев [7]:

1. *Ops API* (сокращенно от *operations* – низкоуровневые операции). Синтаксически эта составляющая близка к классическому *TensorFlow* с *Python*-интерфейсом.

2. *Layers API* (высокоуровневые слои). Это *API* аналогично библиотеке *Keras*, сводящей действия в обучении к минималистичным наборам наиболее распространенных общих команд.

На основе высокоуровневого *API* можно построить собственный интерфейс для взаимодействий пользователя с моделью. Это открывает новые возможности для быстрого создания веб-приложений с захватом данных с сенсоров переносимых устройств (акселерометра, гироскопа, камеры, *GPS* и т.д.) и обучения моделей для подбора релевантного контента, исходя из поведения пользователя [7].

При этом автоматически обеспечивается безопасность, так как данные создаются и хранятся на стороне клиента. Кроме того, *TensorFlow.js* позволяет использовать предобученные модели, что крайне удобно для трудоемких задач, например, таких как распознавание образов [7].

В настоящее время разработчики все чаще и чаще отдают предпочтение фронтенд приложениям перед бэкенд-серверами в качестве сред для машинного обучения. Благодаря *TensorFlow.js*, управление моделями машинного обучения стало возможным в статических *HTML*-документах без настройки сервера или базы данных, позволяя этим сервисам существовать полностью на стороне клиента [5].

Ниже представлен листинг кода для построения модели, предсказывающей координаты точки по заданным параметрам [7].

```
<html>
  <head>
    <!-- Загрузка TensorFlow.js -->
    <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@0.10.0">
</script>
    <script>
      // Определим модель линейной регрессии. Создадим простейшую нейронную сеть
      из одного слоя.
      const model = tf.sequential();
      model.add(tf.layers.dense({units: 1, inputShape: [1]}));

      // Подготовим модель для обучения: обозначим функцию стоимости и
      оптимизатор.
      model.compile({loss: 'meanSquaredError', optimizer: 'sgd'});
      // Создадим некоторые данные для обучения.
      const xs = tf.tensor2d([1, 2, 3, 4], [4, 1]);
      const ys = tf.tensor2d([1, 3, 5, 7], [4, 1]);
      // Обучаем модель по данным.
      model.fit(xs, ys).then(() => {
        // Используем модель для предсказания координаты точки.
        // Откройте этот файл в браузере и зайдите в консоль разработчика.
        model.predict(tf.tensor2d([5], [1, 1])).print();
      });
    </script>
  </head>
  <body>
</body>
</html>
```

### Заключение.

На сегодняшний день, можно смело сказать, что хоть *JavaScript* и не претендует на лавры *Python* в качестве самого оптимального языка для машинного обучения, его актуальность в сфере разработки легковесных и нетребовательных к ресурсам веб-приложений растет в очень крупных масштабах. Его доступность и простота в использовании на фоне роста технологичности и производительности портативных устройств, а также тенденцией к разработке кроссплатформенных приложений очень выгодно выделяются на фоне аналогов. Такие библиотеки машинного обучения, как *TensorFlow.js*, могут заметно преобразить разработку и саму работу веб-приложений, наделяя их полезным и уникальным функционалом.

### Список литературы

- [1] Why use JavaScript for machine learning? [Электронный ресурс]. URL: <https://hub.packtpub.com/should-you-use-javascript-for-machine-learning-and-how-do-you-get-started/> (Дата обращения: 28.03.2023).
- [2] Шолле, Франсуа JavaScript для глубокого обучения: TensorFlow.js / Франсуа Шолле, Эрик Нильсон, Стэн Байлесчи, Шэнкуинг Цэй – СПб.: Питер, 2021. – 576 с.
- [3] JavaScript for Machine Learning? Weighing The Pros and Cons [Электронный ресурс]. URL: <https://javascript.plainenglish.io/javascript-for-machine-learning-weighing-the-pros-and-cons-e7577d0e3f48> (Дата обращения: 28.03.2023).
- [4] Performance Comparison: JavaScript vs. Python for Machine Learning [Электронный ресурс]. URL: <https://dlabs.ai/blog/performance-comparison-javascript-vs-python-for-machine-learning/> (Дата обращения: 28.03.2023).
- [5] The Pros and Cons of Using JavaScript for Machine Learning [Электронный ресурс]. URL: <https://dlabs.ai/blog/the-pros-and-cons-of-using-javascript-for-machine-learning/> (Дата обращения: 28.03.2023).
- [6] TensorFlow.js — библиотека для машинного обучения на JavaScript. [Электронный ресурс]. URL: <https://www.tensorflow.org/js?hl=ru> (Дата обращения: 28.03.2023).

[7] TensorFlow.js: машинное обучение на JavaScript с доставкой в браузер [Электронный ресурс]. URL: <https://proglib.io/p/tensorflow-js> (Дата обращения: 28.03.2023).

## **RELEVANCE OF USING JAVASCRIPT LANGUAGE FOR DEVELOPING MACHINE LEARNING MODELS**

**G.A. Piskun**

*Associate Professor, Department of Information Computer Systems Design, PhD of Technical sciences, Associate Professor*

**V.F. Alekseev**

*Associate Professor, Department of Information Computer Systems Design, PhD of Technical sciences, Associate Professor*

**T.M. Voronko**

*Software Engineer of the Center of Informatization and Innovative Developments of BSUIR, master student of BSUIR*

*Faculty of Computer Engineering*

*Belarusian State University of computer science and Radio Electronics, Republic of Belarus*

*E-mail: voronko232001@gmail.com*

**Abstract.** The applicability and relevance of using the JavaScript programming language for developing machine learning models is analyzed. The main advantages and disadvantages of this method of working with artificial intelligence in comparison with analogues such as Python are considered, and the features of the TensorFlow.js JavaScript library are described, providing an example program for building a model that predicts the coordinates of a point according to given parameters.

As a result of the analysis, it was found that the use of JavaScript can be useful for lightweight and undemanding cross-platform web applications that use machine learning models.

**Keywords:** machine learning, library, artificial intelligence.