IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Overview of Machine Learning using JavaScript

Gopal Sanjay Jawle
Department of Computer Engineering
Government College of Engineering, Yavatmal
gopaljawle02@gmail.com

Sanket Ukanda Gawande
Department of Computer Engineering
Government College of Engineering, Yavatmal
sanketgawande.gcoey@gmail.com

Akshad Pravinkumar Malegaonkar
Department of Computer Engineering
Government College of Engineering, Yavatmal
malegaonkarakshad@gmail.com

Jay Achyutrao Wankhade
Department of Computer Engineering
Government College of Engineering, Yavatmal
wankhadejay13@gmail.com

Prof. Ankush Patil
Department of Computer Engineering
Government College of Engineering, Yavatmal
ankushpatil48@gmail.com

***Abstract*— In modern world the importance of machine learning and artificial intelligence is on a rise. Advancements in technologies have led to the development in the field of machine learning. JavaScript also provides a library for an extensively working model of ml. The libraries namely ML5.js, TensorFlow.js, Brain.js have made it possible to give an essence of machine learning to web technology.**

Keywords*— *ML5.js, Tensorflow.js, Machine Learning, Artificial Intelligence

I. INTRODUCTION

ARTIFICIAL INTELLIGENECE

The research and development of artificial intelligence (AI) has been accelerating during the last few decades. domains including speech recognition, natural language processing, and expert systems Even if there are still big issues that need to be resolved, technologies like recognition, computer vision, robotics, etc. have advanced significantly. The development of new system designs that can utilize all the knowledge, including human skill, available in a particular domain is one of the ways that has contributed to AI's current success. Thus, these knowledge-based systems consider human expertise to enhance their own capabilities. This essay tries to provide an overview of the methodologies, as well as the current possibilities, constraints, and potential future directions of AI.

MACHINE LEARNING

Machine learning is the name of a subfield of artificial intelligence (AI). The primary goal of machine learning is to recognize the structure of data and incorporate it into models that people can understand and use. While machine learning is a subfield of computer science, it differs from traditional computational techniques. Computers use algorithms, which are sets of purposefully designed instructions, to do calculations and address issues in traditional computing. Instead, by using statistical analysis to train on data inputs, computers can provide numbers that fall inside a specified range. With the aid of machine learning, it is possible to programmed computers to create models from sample data in order to automate decision-making based on data inputs. When it comes to machine learning, tasks are frequently divided into broad groups.

Tasks are typically categorized into broad categories in machine learning. Based on how learning is absorbed or how feedback on learning is provided to the system being developed, several categories have been created. Unsupervised learning, which gives the algorithm no labelled data in order to allow it to discover structure within its input data, and supervised learning are two of the most frequently used machine learning techniques. Supervised learning involves training algorithms on example input and output data that has been labelled by humans.

CURRENT LANGUAGES FOR MACHINE LEARNING

Python

Python is the ideal choice for both novice and experienced developers because it can power intricate scripting and internet applications. It was created in 1991 and is a general-purpose programming language. It has long been a favorite among developers because to its simplicity and ease of usage. It works with a wide range of different libraries and frameworks. Python is one of the most in-demand programming languages for machine learning, data analytics, and web development, thus it should come as no surprise that Python engineers are in great demand. Python is a well-liked programming language since it offers a variety of coding alternatives.

Due to their open-source status and scalability, numerous visualization packages and crucial core libraries like sklearn and seaborn are readily available. These strong libraries facilitate coding and facilitate machine learning. Python has support for object-oriented, functional, imperative, and procedural programming models. When it comes to machine learning frameworks, Python coders tend to favor both TensorFlow and Scikit.

Python is suitable for:

- Prototyping
- Scientific computation
- Sentiment analysis
- Natural language processing
- Data science.

R

R is a well-known open-source data visualization-driven language for statistical computation and machine learning. Additionally, it includes a wealth of tools for creating machine learning applications. Because R is simpler to learn than other programming languages, it is used by analysts, statisticians, and data miners. Along with a variety of IDEs and a large selection of tools for managing libraries and creating better graphs, it supports a command-line interface. Data scientists analyse corporate graphs using this graphics-based language, especially in the biomedical sector. R can be used to implement Representational State Machines (RSMs) for classification, regression analysis, and decision tree construction. Due to its statistical and functional characteristics, it has been a dynamic, imperative, and functional language. It may operate on a variety of systems in addition to Windows, Linux, and OS X. According to the Tiobe Index, R is now the 15th most used programming language. You can locate talented R developers for your project with relative ease given the developer community's current level of interest.

JAVASCRIPT LIBRARY FOR MACHINE LEARNING

JavaScript and Java are useful for many machine learning techniques and applications. These object-oriented, data-intensive languages are renowned for their consistency and dependability. There are strong Java frameworks like Weka and Rapid Miner for machine learning techniques like regression and decision trees. Java is additionally a superb choice for enterprise-level software. On the other hand, huge organisations virtually exclusively employ JavaScript in front-end development projects. To be successful in machine learning applications, these technologies rely on the numerous machine learning libraries that are coupled with them. Some of the applications where JavaScript experts use these languages in machine learning are fraud detection, cyber-attack detection, and network security.

SIGNIFICANCE OF JAVASCRIPT OVER OTHER LANGUAGE

Python has recently taken over as the most popular programming language for deep learning and machine learning. Python is primarily used in books and online courses on machine learning and deep learning, either alone or in conjunction with R. Python's broad selection of machine learning and deep learning libraries, as well as its streamlined implementation, scalability, and flexible features, have helped it grow in popularity. However, Python is not the only language available for developing machine learning software. JavaScript is being used increasingly by developers to run machine learning models. There are various benefits to knowing JavaScript machine learning, even though it (yet) cannot fully replace the robust Python machine learning landscape. I will list four.

a. Private machine learning

Client-server topologies are used by most machine learning applications. Wherever the machine learning models are operating, users must transmit their data there. The client-server design clearly has advantages. Through web APIs, developers may run their models on servers and make them accessible to user apps. Developers are now able to use extremely big neural networks that cannot be run on user devices. However, performing the machine learning inference on the user's device is often preferred. Users might not want to submit their emails, private chat messages, or images to the server hosting the machine learning model due to privacy concerns, for instance.

Fortunately, not all applications for machine learning demand expensive servers. In order to run on user devices, many models can be compressed. Additionally, manufacturers of mobile devices are putting chips in their products to facilitate local deep learning inference.

But the issue is that not many user devices come with Python machine learning capability by default. Python is preloaded on MacOS and many Linux distributions, however machine learning libraries must still be installed individually. Python must be manually installed for Windows users. Additionally, Python interpreters are not well supported by mobile operating systems.

On the other hand, all current mobile and desktop browsers natively support JavaScript. This means that the majority of desktop and mobile devices will support JavaScript machine learning apps. Therefore, you can be sure that almost all users will be able to access your machine learning model if it runs on JavaScript code in the browser.

Numerous machine learning libraries for JavaScript are currently available. TensorFlow.js, the JavaScript adaptation of Google's well-known TensorFlow machine learning and deep learning library, serves as an illustration. There is a tons of ready-huge-use JavaScript machine learning examples available if you use your smartphone, tablet, or desktop computer to access the TensorFlow.js demo page. They will not send any data to the cloud; instead, they will run the machine learning models on your device. Additionally, there is no need to set up any additional software. The ML5.js, Synaptic, and Brain.js libraries for JavaScript are additional strong machine learning libraries.

b. Fast and customized ML models

On-device machine learning has further advantages besides privacy. The user experience may be hampered in some apps by the roundtrip delay that results from transmitting data from the device to the server. Users may desire the ability to use their machine learning models even when they are not connected to the internet in other situations. The ability to run JavaScript machine learning models on the user's device is highly helpful in these circumstances.

Model customization is an essential use of JavaScript machine learning. Consider the scenario where you wish to create a machine learning model for text production that can learn each user's preferred language. One alternative would be to train the model using the user's data and save one model per user on the server. As your user base expands, this would increase the strain on your servers and necessitate the storage of potentially sensitive data in the cloud.

An approach would be to build a base model on your server, duplicate it on the user's device, and then utilize JavaScript machine learning tools to refine the model using the user's input.

On the one hand, this would prevent the need to transfer data to the server by keeping it on users' devices. On the other hand, by avoiding sending additional inference and training loads to the cloud, it would free up server resources. Additionally, users' machine learning capabilities would continue be available even if they were not connected to your servers.

c. Easy integration of machine learning in web and mobile applications

The simplicity of integration with mobile applications is another advantage of JavaScript machine learning. Operating system support for Python on mobile devices is still in its infancy. However, there is already a robust selection of cross-platform JavaScript development tools for mobile apps, such Cordova and Ionic.

Because they allow you to create your code only once and distribute it to iOS and Android devices, these technologies have gained a lot of popularity. Cross-platform development tools launch a "WebView" a browser object that can execute JavaScript code and can be embedded in a native application of the target operating system, to make the code compatible across several operating systems. The JavaScript machine learning libraries are supported by these browser objects.

React Native, a well-liked cross-platform mobile app development framework, stands out as an exception because it does not require the WebView to run applications. But because mobile machine learning apps are so common, Google has made TensorFlow.js specifically for React Native available.

You can incorporate your JavaScript machine learning code into a natively developed mobile app by adding your own embedded browser object (WKWebView in iOS, for example).

Other machine learning libraries, such TensorFlow Lite and Core ML, are available for mobile apps. They do, however, necessitate native coding for the mobile platform for which your app is being created. On the other side, JavaScript machine learning has a lot of flexibility. If you have already developed a machine learning application for the browser, you can easily port it with little to no changes to your mobile application.

d. JavaScript machine learning on server

Training the models is one of the main problems with machine learning. This is particularly true for deep learning because this type of learning necessitates costly backpropagation computations spread across multiple epochs. Although you can train deep learning models on consumer devices, if the neural network is big, it can take weeks or months.

For server-side training of machine learning models, Python is more appropriate. To speed up the training process, it may grow and disperse its load among server clusters. Once the model has been trained, it can be delivered compressed and used for inference on user devices. The compatibility of machine learning libraries created in several languages is fortunately very high. For instance, you can save your deep learning model in one of several language-independent formats, such JSON or HDF5, if you train it with TensorFlow or Keras for Python. The user's device can then be loaded with TensorFlow.js or another JavaScript deep learning library using the saved model that was sent to it.

However, it is important to remember that server-side JavaScript machine learning is also progressing. The JavaScript application server engine, Node.js, supports JavaScript machine learning libraries. There is a specific version of TensorFlow.js designed for Node.js servers. The JavaScript code you use to communicate with TensorFlow.js is the same code you would use to communicate with browser-based apps. However, the library uses your server's specialized hardware in the background to quicken training and inference. Although there isn't an official JavaScript implementation for PyTorch, a well-known Python machine learning toolkit, the opensource community has created JavaScript bindings for the library.

Although Node.js machine learning is still relatively young, it is quickly developing due to the increased interest in integrating machine learning capabilities into web and mobile apps. Many web developers who wish to add machine learning to their skill set may choose JavaScript as their preferred alternative as the community and tools continue to improve.

e. Advantages Of JavaScrpit for ml

JavaScript is a great language for use in machine learning because is easy to learn and use, and is has a wide range of libraries and frameworks available. JavaScript also has a large and active community of developers who are constantly creating new tools and libraries for use in machine learning. Additionally, JavaScript is a great for prototyping machine learning algorithms and applications, as it is easy to quickly test and refine ideas

f. Disadvantages Of JavaScrpit for ml

JavaScript is not the best language for machine learning due to its lack of support for numerical computing. Additionally, it is not as easy to debug as other languages, and its lack of support libraries and frameworks makes it difficult to use for complex machine learning tasks.

II. OVERVIEW

DIFFERENT LIBRAY IN JAVASCRITP ARE

Ml5.js

What is ml5.js?

ml5.js is a JavaScript library built on top of Tensorflow.js that provides access to various machine learning and deep learning algorithms within the browser. This library is intended to make machine learning accessible to the huge JavaScript community. Ml5.js aims to make machine learning more approachable to a broad audience of artists, designers, creative coders, and students. The library provides access to machine learning algorithms and models in the browser, building on top of TensorFlow.js with no other external dependencies.Ml5.js is inspired by Processing and P5.js, whose goal is to empower people of all interests and backgrounds to learn how to program and make creative work with code. However, to get started with machine learning, one needs advanced understanding of math and programming. And we would like to make this process easier so that machine learning can be something that everyone can learn, understand, and explore freely.ml5.js uses Tensorflow.js API layer to easily define, train, and test models and it also supports GPU acceleration to enhance

computational efficiency. It has built-in functionalities for other utilities also like visualization, rendering, porting on other platforms which adds to the user's ease of working.

Why use web-based machine learning?
The ability to make machine learning models in the browser is very useful as it can keep the data within the user's browser only. The added bonus of being able to use webcams and microphones directly without using any external libraries as is the case with other languages makes it quite convenient.

Why use ml5.js?
There are other JavaScript libraries for machine learning and Tensorflow.js is quite the popular library then why you should use ml5.js?
Well to begin with ml5.js is built on Tensorflow.js and it adds a layer of API abstraction. This means you do not need to struggle with the low-level nitty-gritty of Tensorflow.js yet you can leverage the same power by using the high-level ml5.js library.
Below are some more points that add to ml5.js credibility.

- Supports acceleration using WebGL and GPUs
- Native support for converting browser I/O streams to model input data structures
- Standardization of model format
- Small model sizes, low-latency, portable model format
- Out of the box Pretrained models for turnkey use.

Tensorflow.js

What is TensorFlow.js?
Tensorflow.js is a JavaScript framework developed by Google that enables developers to train and build machine learning (ML) models to be used in the browser. Developers also have the option of using pre-trained models in lieu of training their own.

History
Before there was Tensorflow.js, there was TensorFlow. Also developed by Google, TensorFlow is an open-source machine learning framework built in C++. It combines several machine learning and deep learning models and algorithms. TensorFlow uses Python to provide a convenient front-end API for building applications while executing applications in C++.
Before TensorFlow.js came into the scene, machine learning in JavaScript was primarily executed by use of an API. The server would deploy the model at the client's request using JavaScript.
In 2017, a Google launched a project called DeepLearn.js, enabled machine learning in JavaScript without the need for an API. Later on in 2018, the DeepLearn.js team was merged with the TensorFlow team to form TensorFlow.js.

What can you do with TensorFlow.js?
If you are developing with TensorFlow.js, here are three workflows you can consider.

- You can import an existing, pre-trained model for inference. If you have an existing TensorFlow or Keras model you've previously trained offline, you can convert into TensorFlow.js format, and load it into the browser for inference.
- You can re-train an imported model. As in the Pac-Man demo above, you can use transfer learning to augment an existing model trained offline using a small amount of data collected in the browser using a technique called Image Retraining. This is one way to train an accurate model quickly, using only a small amount of data.
- Author models directly in browser. You can also use TensorFlow.js to define, train, and run models entirely in the browser using JavaScript and a high-level layer's API. If you are familiar with Keras, the high-level layers API should feel familiar.

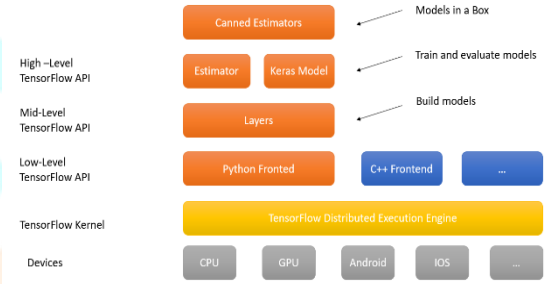

Fig. 1. TensorFlow.js

Brain.js

Brain.js is a GPU accelerated library of neural networks written in JavaScript for browsers and Node.js. It is simple, fast, and easy to use. It provides multiple neural network implementations as different neural nets can be trained to do different things well.
I think this is cool especially because most materials on machine learning are primarily focused around python making the learning curve a bit steep for developers coming from a web background. Brain.JS provides a solution to this. Brain.js is not the first JavaScript focused machine learning library. However, I personally recommend it because it uses JSON objects which eliminates the need for the developer to create tensors and make memory management.
Why Brain.js?
After a basic introduction to brain.js, many developers may have questioned whether is JavaScript good for machine learning and why we need to know about brain.js the simple answer is yes JavaScript is a good fit for machine learning, and many developers use and explore JavaScript space. There are many applications of brain.js already available where we can work on modules and develop an advanced system that predicts future results.
Brain.js is the fastest processing library because of GPU computations and even If you do not have GPU. Beginners can directly jump into brain.js because you don't need neural network knowledge to start with it just basic JavaScript fundamentals are enough to start with brian.js.

III. APPLICATIONS

Ml5.js

ml5.js is machine learning *for the web* in your web browser ml5.js is an open source, friendly high-level interface to TensorFlow.js, a library for handling GPU-accelerated mathematical operations and memory management for machine learning algorithms.
Basically, it is used for machine learning related task using JavaScript as the underlying language.
Ml5 has pretrained modules that contains lots of general things like trees, birds etc. We can customize and feed data related to our task we can train the module as well.

Ml5 uses YOLO for object detection process
YOLO - you look only once (alternative **cocossd**)

Ml5 can be used using its cdn or downloading library code locally
Steps for starting with ml5js

- Adding ml5's official cdn to our web app for accessing it is all functions

- Ml5 has method on it called **object Detector** which is useful in process of object detection
- object Detector takes 3 arguments as input,

1. Model name (yolo or cocossd) Options
2. Call-back function

We can create this object detector instance and save it to any of variable to use it later in process
E.g.: **const** objectDetector= ml5.**objectDetector**('cocossd', {}, modelLoaded);

objectDetector has few methods like detect ()
Detect method
Syntax: objectDetector.**detect**(input, callback);
input: A HTML video or image element or a p5 image or video element. If no input is provided, the default is to use the video given in the constructor.

callback: A function to run once the model has made the prediction. If no call-back is provided, it will return a promise that will be resolved once the model has made a prediction.

Outputs Object: returns an array of objects containing class names, bounding boxes and probabilities.

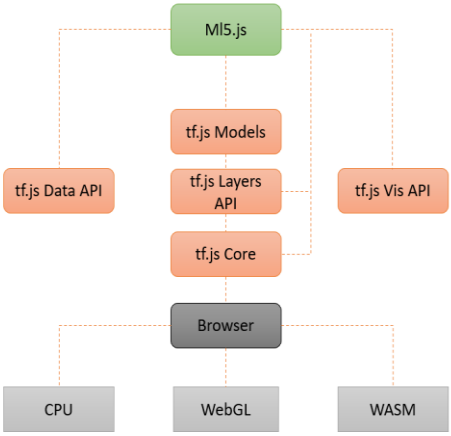

Fig. 2. ML5.js

Brain.js

How to build training data?

In the same index.js we will implement a few lines of code to enable us to train the network.

```
Const _Data = [];
for (let taskName in meetings) {
  const dayOfWeek = meetings[tasks];
  _Data.push({
    input:{[dayOfWeek] : 1},
    input:{[taskName] : 1},
  });
}
```

The above block of code iterates the object of meetings and pushes that value into the training data _Data to be used later to train the network. This _Data takes in an input as the day of the week which is assigned a value of 1. What this means is that whenever a particular day of the week is selected the value automatically is one and all other values will be set to 0 because with Brain.js all undefined values are set to 0 and output as the task.

How to define the neural network and training

As we have seen previously all we have to do is create a new instance of Brain.js like so:

```
const net = new brain.NeuralNetwork({hiddenLayers: [3]});
const stats = net.train(_Data)
console.log(net.run({'Monday': 1}));
```

What the above code displays is the likelihood of each day, so it returns a list of all the days with their probabilities. However, what I want is just a day. So, we create a function:

```
function SpecificDay(dayOfWeek){
  const result = net.run({[dayOfWeek] : 1});
  let highestvalue = 0;
  let highestTaskName = '';
  for(let taskName in result){
    if (result[taskName] > highestvalue){
      highestvalue = result[taskName];
      highestTaskName = taskName;
    }
  }
    return highestTaskName ;
}
```

The above code (also found on code pen) takes the neural network predictions, iterates over them, and then saves the highest value and returns it.

So, if we log this by running:

```
Console.log(SpecificDay("Wednesday"))
```

IV. FUTURE SCOPE

As JavaScript is an already a popular language for a machine learning, researches are being out that how JavaScript will train a model. The ml5 and tenserflow.js it is a new technology to train a model. With the support from a JavaScript community is working on Tenserflow.js and ml5.js. Every month a new version is being released on this technology.
JavaScript library helps to JavaScript developers to build and deploy ml model within the client-side applications, it runs models in the browser. And the community will get help in user-friendly machine learning models to deploy in web applications.

V. CONCLUSION

In this paper we have briefly discussed about the usability of JavaScript in the domain of Machine Learning. We have also elaborated the use of ML5.js, TensorFlow.js, Brain.js to train and test numerous machine learning models.

ML5.js is a JavaScript library built on top of Tensorflow.js that provides access to various machine learning and deep learning algorithms within the browser. It also supports GPU acceleration to enhance computational efficiency. The ability to make machine learning models in the browser is very useful as it can keep the data within the user's browser only. TensorFlow is an open-source machine learning framework built in C. You can use transfer learning to augment an existing model trained offline using a small amount of data collected in the browser using a technique called Image Retraining. Brain.js is a GPU accelerated library of neural networks written in JavaScript for browsers and Node.js. Brain.js is not the first JavaScript focused machine learning library.

VI. REFERENCE

[1]A. Verma, C. Kapoor, A. Sharma and B. Mishra, "Web Application Implementation with Machine Learning," *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, 2021, pp. 423-428, doi: 10.1109/ICIEM51511.2021.9445368.

[2] S. Ndichu, S. Ozawa, T. Misu and K. Okada, "A Machine Learning Approach to Malicious JavaScript Detection using Fixed Length Vector Representation," 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1-8, doi: 10.1109/IJCNN.2018.8489414.

[3] S. Tilkov and S. Vinoski, ‘‘Node.js: Using JavaScript to build highperformance network programs,’’ IEEE Internet Comput., vol. 14, no. 6, pp. 80–83, Nov./Dec. 2010

[4] J. Á. Morell, A. Camero and E. Alba, "JSDoop and TensorFlow.js: Volunteer Distributed Web Browser-Based Neural Network Training," in IEEE Access, vol. 7, pp. 158671-158684, 2019, doi: 10.1109/ACCESS.2019.2950287.

[5] X. Pournaras and D. A. Koutsomitropoulos, "Deep Learning on the Web: State-of-the-art Object Detection using Web-based Client-side Frameworks," 2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA, 2020, pp. 1-8, doi: 10.1109/IISA50023.2020.9284358.

[6] S. Wang, Q. Duan, K. -S. Wong and C. A. Ardagna, "Machine learning for mobile edge computing," in *China Communications*, vol. 18, no. 11, pp. iii-v, Nov. 2021, doi: 10.23919/JCC.2021.9631177.

[7] Ferenc, R., Hegedus, P., Gyimesi, P., Antal, G., Ban, D., & Gyimothy, T. (2019). Challenging Machine Learning Algorithms in Predicting Vulnerable JavaScript Functions. 2019 IEEE/ACM 7th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE). doi:10.1109/raise.2019.00010

[8] Pupo, A. L. S., Nicolay, J., Efthymiadis, K., Nowe, A., Roover, C. D., & Boix, E. G. (2019). GUARDIAML: Machine Learning-Assisted Dynamic Information Flow Control. 2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER). doi:10.1109/saner.2019.8667979