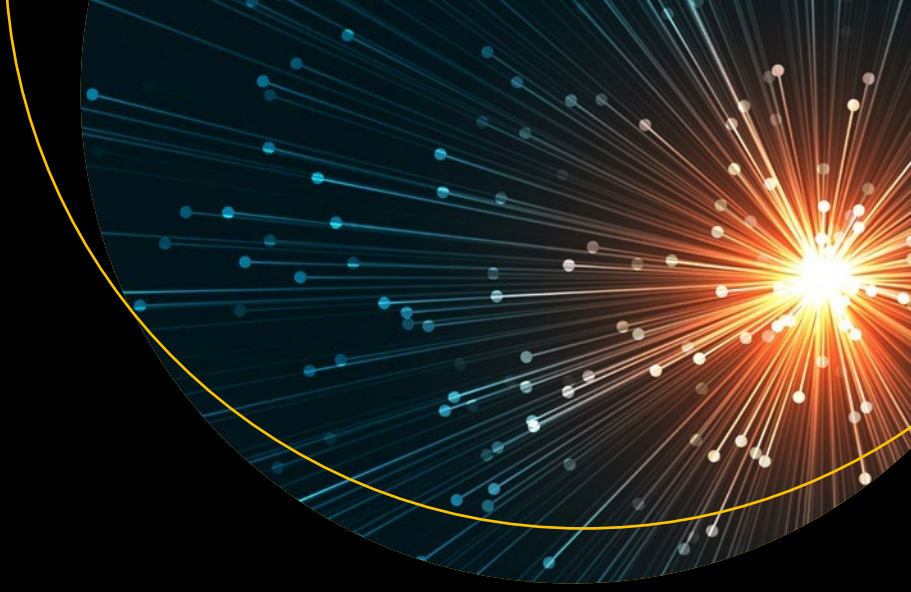

Beginning Machine Learning in the Browser

Quick-start Guide to Gait Analysis with JavaScript and TensorFlow.js

Nagender Kumar Suryadevara

Apress®

Beginning Machine Learning in the Browser

Quick-start Guide to Gait Analysis with JavaScript and TensorFlow.js

Nagender Kumar Suryadevara

Apress®

Beginning Machine Learning in the Browser: Quick-start Guide to Gait Analysis with JavaScript and TensorFlow.js

Nagender Kumar Suryadevara
School of Computer and Information Sciences,
University of Hyderabad, Hyderabad, Telangana, India

ISBN-13 (pbk): 978-1-4842-6842-1 ISBN-13 (electronic): 978-1-4842-6843-8
https://doi.org/10.1007/978-1-4842-6843-8

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Aaron Black
Development Editor: James Markham
Coordinating Editor: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 NY Plazar, New York, NY 10014. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-6842-1. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Table of Contents

About the Author

Dr. Nagender Kumar Suryadevara received his Ph.D. degree from the School of Engineering and Advanced Technology, Massey University, New Zealand, in 2014. He is an Associate Professor at the School of Computer and Information Sciences, University of Hyderabad, India. His research interests include wireless sensor networks, the Internet of Things, and time-series data mining. He has authored two books, edited two books, and published more than 50 papers in various international journals, conferences, and book chapters. He has delivered numerous presentations, including keynote, tutorial, and special lectures. He is a senior member of IEEE. Dr. Suryadevara is passionate about development possibilities for great AI-based products in resource-constrained computing environments. Google Scholar citations:h-index:19,i10-index:30.

https://scholar.google.com/citations?user=S28OdGMAAAAJ&hl=en

About the Technical Reviewer

Vishwesh Ravi Shrimali graduated from BITS Pilani, where he studied mechanical engineering, in 2018. Since then, he has worked with BigVision LLC on deep learning and computer vision and was involved in creating official OpenCV AI courses. Currently, he is working at Mercedes Benz Research and Development India Pvt. Ltd. He has a keen interest in programming and AI and has applied that interest in mechanical engineering projects. He has also written multiple blogs on OpenCV and deep learning on LearnOpenCV, a leading blog on computer vision. He has also authored *Machine Learning for OpenCV* (2nd edition) by Packt. When he is not writing blogs or working on projects, he likes to go on long walks or play his acoustic guitar.

Acknowledgments

This journey would not have been possible without the support of my family, professors, mentors, and friends. I am especially grateful to my parents, who supported me emotionally. To my family, thank you for encouraging me in all of my pursuits and inspiring me to follow my dreams.

I want to express my sincere gratitude and heartfelt thanks to **Vishwesh Ravi Shrimali** for allocating some time and taking responsibility for reviewing the book chapters and providing valuable comments and helpful suggestions.

Thanks to everyone in the publication team, especially **Aaron Black**, who helped me in getting the book content into great shape, **James Markham**, who took great pains in grooming the book, and **Jessica Vakili,** who made sure the book writing process went smoothly and on time.

I am indebted to many of my students and colleagues who were involved with various projects over several years, and some of their works have been used in this book. I would especially like to give credit to one of my students, Ashish Gupta, for helping me in executing the programs in Chapter 5.

Preface

In recent times, artificial intelligence (AI) and machine learning (ML) techniques have been widely used in many applications, such as monitoring environmental parameters, monitoring, and control of industrial situations, intelligent transportation, structural health monitoring, health care, and so on. The advancement of electronics, embedded control, smart sensing, networking, and communication has made it possible to develop low-cost smart systems. Although there are smart systems, the computing capabilities are minimal, and hence they are considered to be resource-constrained computing devices (e.g., mobile phones, smart watches, and mini electronic gadgets).

Applying smart strategies involving complex mathematical operations of AI and ML methods on resource-constrained computing devices and browsers is challenging. The advancements in Internet technologies, primarily JavaScript skills, have made it possible to execute AI/ML models in the browsers and resource computing devices. There are many other publications on AI, ML, and JavaScript, but this book provides beneficial information and practical knowledge to develop intelligent methods/models from scratch and deploy them on browsers and resource-constrained computing devices.

The complete book is divided into six chapters:

Chapter 1 describes the fundamentals of web development. This chapter provides a short description of designing and developing web applications using web building blocks. For developing an AI/ML model and running on the browser or resource-constrained computing device, this chapter's practical knowledge is essential. For a beginner in any field

of study and interested in developing web apps, this chapter provides the necessary practical skills to better realize web applications.

Chapter 2 delivers the steps to be performed and the necessary JavaScript libraries to be considered for processing the data at the computer's browser application level. The latest JavaScript libraries (p5.js and ml5.js) that help build the AI/ML models with practical steps are covered.

Chapter 3 introduces the human pose estimation application as an example that enables the reader to understand how an AI/ML model involving complex mathematical operations can be used to run in the browser. The stepwise procedure teaches you how to implement ML methods to estimate an individual's poses.

Chapter 4 covers the open-source JavaScript library TensorFlow.js, which will be useful for building and deploying AI/ML models in the browser. The architecture of the TensorFlow.js, including its accelerators that support massive data processing at the browser, is explained. Practical examples of executing a neural network model for useful classification tasks are elucidated using the TensorFlow.js library.

Chapter 5 examines how to determine gait parameters by applying AI/ML methods along with the JavaScript libraries in the web browser application. The chapter walks you through the basics of gait analysis and expands on the observational method considered in determining the vital parameters for analysis using the AI in the browser.

Chapter 6 provides a few more advanced applications to run on the browser by applying the AI/ML methods. This chapter encourages the reader to think about the advancements possible when running AI/ML models in the browser.

I hope that you enjoy reading the book. If you need any help whatsoever with the practicals, please feel free to contact me.

Dr. S. Nagender Kumar Suryadevara
Associate Professor, School of Computer and Information Sciences,
University of Hyderabad, India.

CHAPTER 1

Web Development

This chapter introduces you to the fundamentals of machine learning (ML) and provides a practical primer to web design and development for complete beginners. Topics covered in this chapter include the following:

- Hypertext Markup Language (HTML)
- Cascading Style Sheets (CSS)
- JavaScript (JS)
- Document Object Model (DOM)
- jQuery

These building blocks of web development enable you to implement rich user functionalities into your web design.

Machine Learning Overview

Machine learning, a subset of artificial intelligence (AI), aims to enable computers to learn without interacting with specific programs. ML enables computers to develop programs that can access data and use it to learn for themselves (and thus perform like a human).

N. K. Suryadevara, *Beginning Machine Learning in the Browser*,
https://doi.org/10.1007/978-1-4842-6843-8_1

Arthur Samuel, who believed that computers could learn without specific programs, popularized the term *machine learning* in 1959. In 1997, Tom Mitchell further clarified the concept of ML, stating that a computer could learn from some relative measure involving past performance while processing some task, thus giving some *experience* to the computer.

Today, electronics of all kinds are outfitted with cutting-edge, high-sensitivity sensors. Further, Internet connectivity allows for communication among gadgets (things) for better environment-condition monitoring. Accordingly, the massive amount of data generated from these gadgets drives the Internet of Things (IoT) concept. Using AI and ML strategies, the broad information gathered can be processed, scaled, ordered, and used to predict events.

In conventional ML approaches, data is sent to and handled through a central server, which experiences communication overhead, latency, protection loss, and security issues. To overcome these difficulties, inferences from the data collected in the IoT realm can be made by deploying better ML techniques near the data origin using, for instance, browser-environment capabilities. Exploiting ML strategies on resource-constrained computing devices through a browser helps respective entities to make better decisions in real time for enhanced functionality.

The tremendous computational demands of current AI strategies and the development of ever-increasing numbers of AI-enhanced applications forecast more data-processing problems. After all, computer-based intelligence systems features are more demanding as they seek to reduce resource utilization, to quicken resource accessibility, and to exploit resource utilization for precision.

Software developers and engineers can now more effectively leverage AI to conceptualize exceptionally responsive applications that respond to user-sourced information in real time, such as voice or facial recognition. They can also make smarter applications that can learn from user behavior.

Computer-based intelligence enables us to automate applications to incorporate substantive proposals, to respond to voice requests or physical motions, to use mobile phone cameras to recognize items or places, and to figure out how to help users with day-by-day activities.

In the past, many of the best ML and deep learning (DL) systems required familiarity with Python and its related library system. Production of ML models required unique reasoning equipment and programming tools, such as NVIDIA graphical processing units (GPUs) and CUDA. Now, however, incorporating ML into JavaScript (JS) applications often involves deploying the ML part on remote cloud systems, such as Amazon Web Services (AWS), and getting the model to run on the local system via application programming interface (API) calls. This nonlocal, back-end centered methodology has likely kept many web engineers from taking advantage of the abundant prospects AI offers to front-end improvement.

The main advantage of running AI strategies on users' local devices (i.e., near the data-origin source) is that the information never leaves the user's device. This point is critically significant because users rightfully worry about their data privacy, especially in the wake of well-publicized and embarrassing information leaks and security breaches.

With the help of TensorFlow.js software tools, developers/users can exploit AI without sending their information over a system that potentially makes it available to an outsider. These tools make it simpler to develop secure applications that comply with information security guidelines, such as healthcare applications that read wearable clinical sensors. The tools also make AI program augmentation possible, thus allowing upgrades while shielding user conduct/information.

Integrating JS programming features with AI strategies in a simple interface can lead to more straightforward access to rich sensor information from IoT devices. User behavior can be modeled based on interactivity with device information sources such as voice or webcams. Because similar programming code can run on, for example, mobile

phones utilizing accelerometers, gyroscopes, and Global Positioning System (GPS), integrating AI computational capabilities into user devices themselves can prove highly beneficial.

Web Communication

Figure 1-1 shows the big-picture web basics for AI applications that run on user devices.

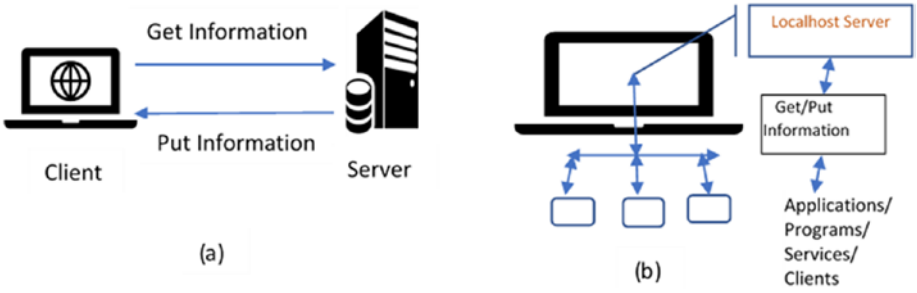

Figure 1-1. *Web communication through the Internet and localhost*

The three web development essentials are as follows:

- Client (web browsers, used to surf the web)
- Server systems (used to supply information to the browsers)
- Computer networks (used to support browser-server communication)

The web activity shown in Figure 1-1(a) illustrates the internetworking principle, where communication between the client and server is done through protocols such as the Internet Protocol (IP), Transmission Control Protocol (TCP), Hypertext Transfer Protocol (HTTP), and the File Transfer Protocol (FTP). Figure 1-1(b) shows that communication between

the client (browser/services/applications) and the server (localhost) happens locally and provides the required information to the respective applications (client/browser/services).

The following terms relate to the communication:

- *World Wide Web (WWW or web)*: A system of interlinked, hypertext documents that runs over the Internet. There are two types of software: client and server. A system that wants to access the information provided by servers must run client software (e.g., a web browser), and an Internet-connected computer that wants to provide information to others must run server software. The client and server applications communicate over the Internet by following a protocol built on TCP/IP (i.e., HTTP) (Figure 1-2).

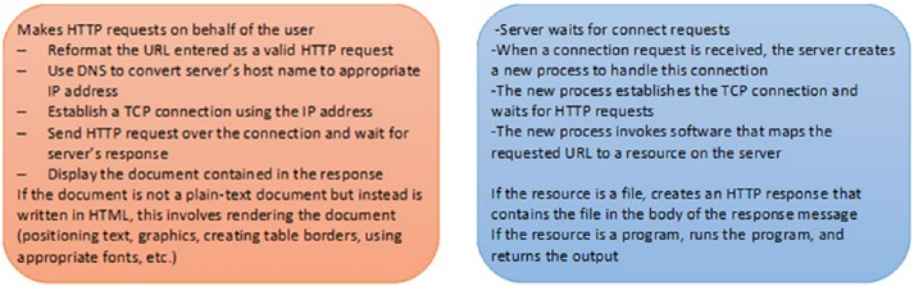

Figure 1-2. *The communication between the web client (browser) and the web server*

- *Hypertext*: An information format that enables one to move from one part of a document to another or from one document to another through hyperlinks.

- *Uniform Resource Locator (URL)*: Unique identifiers used to locate a particular resource on the network.
- *Markup language*: Defines the structure and content of hypertext documents.

Organizing the Web with HTML

To design and develop web pages, you want to be thoroughly familiar with Hypertext Markup Language (HTML). HTML enables you to define a web page's structure, including sections, lists, headings, connection points, pictures, mixed-media players, and more.

HTML is not a programming language. It is a markup language that tells Internet browsers how to structure web pages that a user visits. HTML consists of various components that you use to manipulate substantive page elements to show in a specific way. Encasing labels, for instance, can turn content into a hyperlink that associates with another page or can be used to emphasize words/terms.

Web Development Using IDEs/Editors

The difference between an integrated development environment (IDE) and an editor (text) for web development is that an IDE does everything from fundamental content management to advanced development that cannot be done with a text editor.

For example, editors such as ***Sublime***, ***Notepad++***, and ***Atom*** can be used with HTML and Cascading Style Sheets (CSS) when writing the code for web page design. These editors include many good features (e.g., language structures that include adaptable interfaces and comprehensive navigation tools for web developers who want enhanced application capabilities).

For instance, a web developer may require a debugger and a compiler to develop web applications effectively. Figure 1-3 shows the programming environments of these three editors.

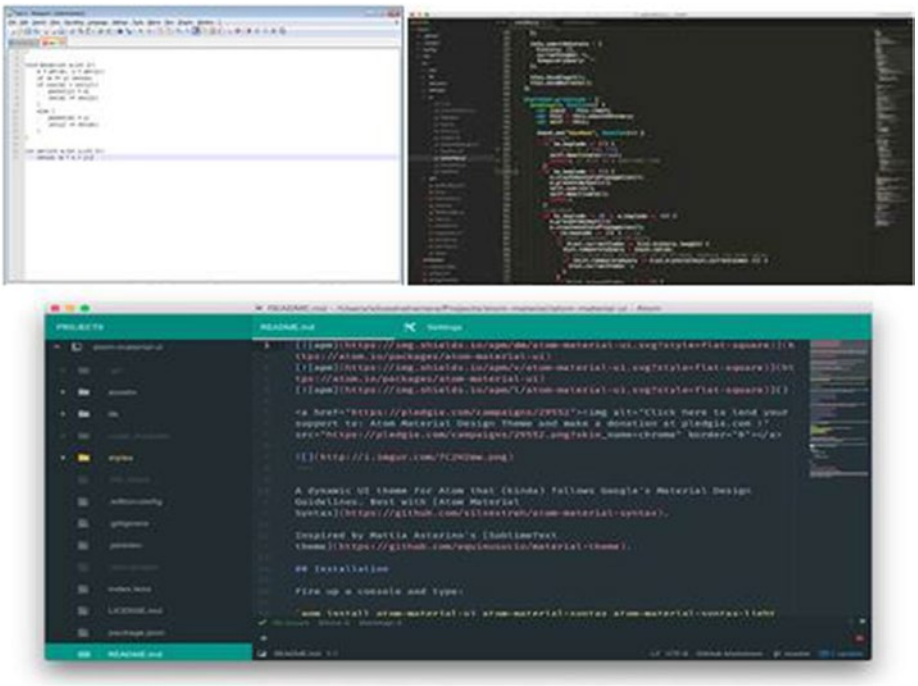

Figure 1-3. *Notepad++, Sublime, and Atom editor environments*

With the best IDEs, however, you have less to worry about. They often include comprehensive development tools in one application, including for automation, testing, and forecasting. Mainly, they provide web developers the necessary support to transform code into a working application. Here are some of the more popular IDEs:

- *Visual Studio Code*: Visual Studio Code is perhaps the best JavaScript IDE for Windows, Mac, and Linux platforms. In addition to supporting JS functionality,

it also supports Node.js and TypeScript features, and it includes a system of extensions for different programming dialects, including C++, C#, Python, and PHP. Visual Studio Code makes for programmer-friendly operations with excellent syntax features and autocomplete with IntelliSense that responds to myriad factors, word definitions, and imported modules.

- *NetBeans*: NetBeans is one of the best web development IDEs because it enables you to create a neat and versatile work area and develop web applications quickly. It also works well with JS, HTML5, PHP, and C/C++. It is a free JS IDE and a great HTML5 IDE for everyday use. This IDE allows you to review code for errors and lets you automatically fix syntax if necessary (including for Java 8 features such as lambda expressions).
- *PyCharm*: PyCharm is not the best free JS IDE. However, the paid Professional Edition is worth considering if you are looking for a solid web development IDE for Python.
- *IntelliJ IDEA*: IntelliJ IDEA is an excellent web development IDE. A free version is available, but if you want all the JS features it offers, consider the paid Ultimate Edition. IntelliJ IDEA can save you time and energy in web development, and it is an excellent CSS IDE. Note, as well, that it supports a wide range of programming dialects.