Mindluster Platform

About the course Machine Learning USING JAVASCRIPT

You can run JavaScript machine learning libraries on Node. js, the JavaScript application server engine. TensorFlow. js has a special version that is suited for servers running Node. Thanks to new AI libraries, JavaScript developers can now build machine learning and deep learning applications without Python or R. ... With AI, JavaScript developers can make more intelligent web applications. Even though there are many different skills to learn in machine learning it is possible for you to self-teach yourself machine learning. There are many courses available now that will take you from having no knowledge of machine learning to being able to understand and implement the ml algorithms yourself. js is a JavaScript library created by Google as an open-source framework for training and using machine learning models in the browser. In short, the framework helps JavaScript developers build and deploy ML models within client-side applications. While JavaScript is not a replacement for the rich Python machine learning landscape (yet), there are several good reasons to have JavaScript machine learning skills. Here are four. Private machine learning Most machine learning applications rely on client-server architectures. Users must send their data where the machine learning models are running. There are clear benefits to the client-server architecture. Developers can run their models on servers and make them available to user applications through web APIs. This makes it possible for developers to use very large neural networks that canâ€™t run on user devices. In many cases, however, it is preferable to perform the machine learning inference on the userâ€™s device. For instance, due to privacy issues, users may not want to send their photos, private chat messages, and emails to the server where the machine learning model is running. Fortunately, not all machine learning applications require expensive servers. Many models can be compressed to run on user devices. And mobile device manufacturers are equipping their devices with chips to support local deep learning inference. But the problem is that Python machine learning is not supported by default on many user devices. MacOS and most versions of Linux come with Python preinstalled, but you still have to install machine learning libraries separately. Windows users must install Python manually. And mobile operating systems have very poor support for Python interpreters. JavaScript, on the other hand, is natively supported by all modern mobile and desktop browsers. This means JavaScript machine learning applications are guaranteed to run on most desktop and mobile devices. Therefore, if your machine learning model runs on JavaScript code in the browser, you can rest assured that it will be accessible to nearly all users. There are already several JavaScript machine learning libraries. An example is TensorFlow.js, the JavaScript version of Googleâ€™s famous TensorFlow machine learning and deep learning library. If you head to the TensorFlow.js demo page with your smartphone, tablet, or desktop computer, youâ€™ll find plenty of ready examples using JavaScript machine learning. They will run the machine learning models on your device without sending any data to the cloud. And you donâ€™t need to install any additional software. Other powerful JavaScript machine learning libraries include ML5.js, Synaptic, and Brain.js. Fast and customized ML models Privacy is not the only benefit of on-device machine learning. In some applications, the roundtrip of sending data from the device to server can cause a delay that will hamper the user experience. In other settings, users might want to be able to run their machine learning models even when they donâ€™t have an internet connection. In these cases, having JavaScript machine learning models that run on the userâ€™s device can come in very handy. Another important use for JavaScript machine learning is model customization. For example, suppose you want to develop a text generation machine learning model that adapts to the language preferences of each user. One solution would be to store one model per user on the server and train it on the userâ€™s data. This would put extra load on your servers as your users grow and it would also require you store potentially sensitive data in the cloud. An alternative would be to create a base model on your server, create a copy on the userâ€™s device, and finetune the model with the userâ€™s data using JavaScript machine learning libraries. On the one hand, this would keep data on usersâ€™ devices and obviate the need to send them to the server. On the other hand, it would free up the resources of the server by avoiding to send extra inference and training loads to the cloud. And users would still be able to use their machine learning capabilities even when theyâ€™re disconnected from your servers. Easy integration of machine learning in web and mobile applications Another benefit of JavaScript machine learning is easy integration with mobile applications. Python support in mobile operating systems is still in the preliminary stages. But there is already a rich set of cross-platform JavaScript mobile app development tools such as Cordova and Ionic. These tools have become very popular because they enable you to write your code once and deploy it for iOS and Android devices. To make the code compatible across different operating systems, cross-platform development tools launch a â€œwebview,â€ a browser object that can run JavaScript code and can be embedded in a native application of the target operating system. These browser objects support JavaScript machine learning libraries. One exception is React Native, a popular cross-platform mobile app development framework that does not rely on webview to run applications. However, given the popularity of mobile machine learning applications, Google has released a special version of TensorFlow.js for React Native. If you have written your mobile app in native code and want to integrate your JavaScript machine learning code, you can add your own embedded browser object (e.g., WKWebView in iOS) to your app. There are other machine learning libraries for mobile applications, such as TensorFlow Lite and Core ML. However, they require native coding in the mobile platform you are developing your app for. JavaScript machine learning, on the other hand, is very versatile. If you have already implemented a version of your machine learning application for the browser, you can easily port it to your mobile application with little or no changes. JavaScript machine learning on server One of the main challenges of machine learning is training the models. This is especially true for deep learning, where learning requires expensive backpropagation computations over several epochs. While you can train deep learning models on user devices, it could take weeks or months if the neural network is large. Python is better suited for server-side training of machine learning models. It can scale and distribute its load on server clusters to accelerate the training process. Once the model is trained, you can compress it and deliver it on user devices for inference. Fortunately, machine learning libraries written

in different languages are highly compatible. For instance, if you train your deep learning model with TensorFlow or Keras for Python, you can save it in one of several language-independent formats such as JSON or HDF5. You can then send the saved model to the userâ€™s device and load it with TensorFlow.js or another JavaScript deep learning library. But it is worth noting that server-side JavaScript machine learning is also maturing. You can run JavaScript machine learning libraries on Node.js, the JavaScript application server engine. TensorFlow.js has a special version that is suited for servers running Node.js. The JavaScript code you use to interact with TensorFlow.js is the same you would use for applications running in the browser. But in the background, the library makes use of the special hardware of your server to speed up training and inference. PyTorch, another popular Python machine learning library, doesnâ€™t yet have an official JavaScript implementation, but the open source community has developed JavaScript bindings for the library. Machine learning with Node.js is fairly new, but it is fast evolving because there is growing interest in adding machine learning capabilities to web and mobile applications. As the JavaScript machine learning community continues to grow and the tools continue to mature, it might become a go-to option for many web developers who want to add machine learning to their skillset.

Programming Category's Courses

Course Lesson(5)

Lesson 1 : **Machine Learning Tutorial for Beginners USING JAVASCRIPT**

Lesson 2 : **Machine Learning Tutorial Making a recommendation engine IN THE BROWSER**

Lesson 3 : **Neural Networks Explained Machine Learning Tutorial for Beginners**

Lesson 4 : **Neural Networks Explained Pt 2 Machine Learning Tutorial for Beginners**

Lesson 5 : **TensorFlow JS Tutorial Build a neural network with TensorFlow for Beginners**

Related courses

React Projects Workshops

Node js back end JavaScript

Python programming language

Social Network Theme UI With Sass

Full Stack React Django

Build a Node js App With Sequelize

for Business Contact
business@mindluster.com