WebAssembly powered Machine Learning: Ndarrays

Bhonsle, Archit Patil, Ved Valkunde, Tanvi

May 15, 2021

Abstract

The project aims to provide a fast, easy-to-use and intuitive machine learning library that runs in the browser. It will encompass the entire machine learning process fueled by WebAssembly. This report will go over the specifics of the implementation of the ndarrays library and the development of its API.

Contents

Chapter 1

Introduction

1.1 Background

In today's day and age, machine learning has become increasingly an integral part of our businesses and lives. However, the current machine learning landscape is dominated by Python and its ML ecosystem consisting of Numpy, Pandas, Matplotlib and Sklearn. The most common way we interface with machine learning is through our browsers. Every time Facebook correctly tags someone in the photos we upload, or how YouTube decides to recommend us videos a machine learning model is running behind the scenes to make those predictions. The traditional way to do this is to use a server that takes in all the data and produces the output and sends it back. For complex models, this makes a lot of sense but for simple predictions, it is quite a bit of overhead when the calculation can be done on the device itself. The problem with using python for this is that they need a python run-time on the system which is not available on browsers. This project aims to build a machine learning library that runs in the browser powered by WebAssembly. This paper will focus on the building of the ndarrays implementation which is a fundamental component of any machine learning library as all machine learning operations can be reduced to matrix operations.

1.2 Motivation

The most widely used machine learning library is TensorFlow.js, the JavaScript equivalent of TensorFlow. Its main focus is heavy OpenGL accelerated deep learning in the browser. Its competitors like keras.js, synaptic.js, brain.js, mind all focus on neural networks. The problem with deep learning is it is computationally expensive and more often than not we need to reach for

GPU acceleration. WebAssembly is a new technology the frees us from the shackles of JavaScript and allows us to write code that runs at a native speed that runs in the browser. We felt a need to create a library that performs basic machine learning tasks like linear regression, k-mean clustering, support vector machines etc. And instead of just building the machine learning library we aim to build a complete machine learning framework powered by WebAssembly.

1.3 Problem Definition

Since the most common ndarray implementation used for machine learning is numpy which is a python library and thus needs a python run-time to use. Because of this, we can't use it in the browser. JavaScript being a dynamic scripting language has a speed penalty of not having static types and being interpreted at run-time. WebAssembly which became a World Wide Web Consortium recommendation since 5 December 2019. It allows us to write code that runs at almost native speed. The aim of this project is to build a fast, easy-to-use ndarrays library that runs in the browser leveraging WebAssembly.

1.4 Scope

The project will only implement core machine learning algorithms, mainly, regression, classification and clustering algorithms. The advantage of sticking to these is that not only are they quite powerful but also less computationally expensive and need fewer data compared to deep learning algorithms. We can also avoid GPU acceleration which however may be introduced in the later stages of the project utilizing WebGPU. The project also utilizes WebAssembly which is a part of the core web alongside HTML, CSS and JS since December 2019. The ndarrays module will implement various matrix operations on the core data types.

1.5 Limitations

1. Since we aren't using GPU acceleration our matrix operations may be slower for large matrices than libraries that do use hardware acceleration.

2. The project is being built on bleeding edge technologies and so the browser support may be low for now.