Table of Contents

J-AIL

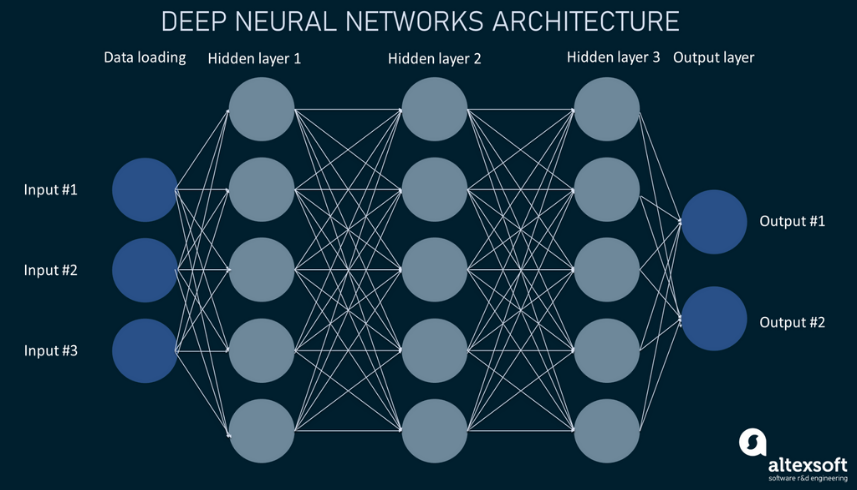
Just-in-time Artificial Intelligence Library

Executive Summary 2

Introduction 3

J-AIL

Executive Summary

Python is the leading programming language for AI development. However, its performance limitations hinder large-scale computations. Top end AI models make billions of computations per operation, and Python’s performance becomes a bottleneck. J-AIL aims to solve this bottle neck by leveraging the computational speed of C++.

In addition to performance, J-AIL aims to solve key issues found in existing AI development libraries. Many of them are bloated, complex and too generalized. By building a custom library, we can create a high-performance and lightweight toolkit tailored to our specific needs. With this tool our development team will be able to push the boundaries of AI and contribute to the future of technology.

Introduction

In the current age of AI, it is imperative to have a way for developers to make quick and efficient AI models, J-AIL will be the means in which they can do that. J-AIL will be an artificial intelligence framework that focuses on speed and simplicity. Because it is a programming framework that means it will only be used by the developers of the team. But in the end, developers using J-AIL will be capable of making high end AI models.

J-AIL will be similar to existing AI development libraries such as JAX and TensorFlow, the main difference will be how lightweight it is. Developing our own framework instead of using existing frameworks can be beneficial for a few reasons. First the developers won’t have to read through confusing and convoluted documentation. For example, TensorFlow, though powerful, has many pages of documentation that developers would have to sift through in order to find meaningful information (TensorFlow, 2024). Second, the lightweight framework will help in keeping code simple and maintainable by only providing necessary functionality. Both of these will assist in rapid development of AI models.

One of the main reasons for developing J-AIL is that it’s written in C++ as opposed to Python. “C++ can be up to 100 times faster than Python” (capaciteam, n.d.). This is important because of the billions of computations that AI models need in order to effectively train to be accurate. Although other frameworks like JAX are also written in C++, their Python-based application programing interface introduces an unnecessary layer of abstraction that complicates the development process (JAX, 2024).

In conclusion, J-AIL intends to fill a gap in AI development frameworks by offering a lightweight, high-performance framework. Its lightweight nature will reduce code complexity and give developers greater control over their AI models, leading to faster development cycles, improved performance, and a more efficient AI development process overall.

***References***

JAX. (2024). *Foreign Function Interface (FFI).* JAX. <https://docs.jax.dev/en/latest/ffi.html>

TensorFlow. (2024). *All Symbols in TensorFlow 2.* TensorFlow. <https://www.tensorflow.org/api_docs/python/tf/all_symbols>

Capaciteam. (n.d.) *C++ vs. Python: All You Need to Know.* Capaciteam. <https://capaciteam.com/c-plus-plus-vs-python/>