Language Bindings for TensorFlow

Irene Wang

Abstract

TensorFlow is a software library for numerical computation using data flow graphs. Hypothetically, we are trying to run an application server proxy herd on a large set of virtual machines, and TensorFlow is being used for machine learning algorithms. TensorFlow usually uses a high level language like Python to set up the graphs, and then performs actual computation using C++. A prototype using Python exists-- however, the performance is being bottlenecked by the Python code, when typically TensorFlow applications are bottlenecked inside C++ code. One potential way that performance could be sped up would be to use a different language instead of Python. In this paper, I will discuss the advantages and disadvantages of potentially using Java, OCaml, and Kotlin instead of Python to improve performance.

1. TensorFlow in Python

Python was the first language to be used for TensorFlow; therefore, it supports the most TensorFlow features. However, it is unsurprising that performance issues are coming up with Python. It is an interpreted language, as well as a dynamically typed language, features that generally make it slower, as these cause much more overhead. Python memory management consists of a private heap that contains all objects and data structures. Its method of garbage collecting uses reference counting and gets rid of objects that have no references to them. This method is slower than some other garbage collection methods, such as the mark-and-sweep garbage collection used in Java.

Generally, the slowness of Python is not as much of an issue because much more time is spent running the C++ code, as machine-learning is usually very computationally heavy. However, since we handle many small queries using tiny models, more of the time is spent running Python code to set up the models rather than C++ code-- leading to slower performance.

Python has the asyncio library which allows developers to write code that runs asynchronously, but does not have multithreading capabilities, so it cannot take full advantage of the speed that comes with executing threads in parallel.

2. Language Bindings for TensorFlow

More of the functionality of TensorFlow is gradually being moved into its core, which is implemented in C++, and support for new languages is built on top of the C API. However, not all functions available in the Python API are available in the C API currently.

3. Java

Java is a class-based and object oriented language that is compiled rather than interpreted, though it still has some properties of interpreted languages. Java has static type checking, so type checking is executed at compile-time rather than at runtime, making it faster than a dynamically type checked language such as Python. Garbage collection is done automatically using the mark-and-sweep method, which is faster than the garbage collection method of Python.

Another feature that Java has is its support for running code in parallel. Allowing different threads to execute tasks at the same

time can greatly speed up a program as compared to using asynchronous libraries. Java also has libraries to support TCP protocols.

Since Java has features of both interpreted and compiled languages, it is very portable due to the Java Virtual Machine, while it still offers good performance.

4. OCaml

OCaml is the main implementation of the Caml language. OCaml is primarily a functional language, but it also allows for object-oriented programming. It uses static typing, preventing type mismatches during runtime and allowing for faster performance than dynamically typed languages such as Python.

OCaml has an async library that supports asynchronous code for event-driven servers, but it does not support multithreading. Similar to Python, it has a global interpreter lock that prevents multiple threads from running code at the same time.

A major issue with OCaml is that it is a functional programming language, making it difficult to compile to efficient machine language code. In addition, the style of functional programming is very different from and less intuitive than imperative programming (such as Java), giving it a steeper learning curve.

Finally, TensorFlow does not maintain the bindings for OCaml; they are created by a third-party.

5. Kotlin

Kotlin is a very new language, as it was released in 2011. Kotlin is also a statically typed language, which again offers a performance advantage over Python. It has many similarities to Java, and the developers claim that a Kotlin application runs as fast or even faster than Java applications. Kotlin is also more concise, with an estimated 40% reduction in the number of lines of code necessary.

Kotlin provides language-level support of coroutines for asynchronous programming, so it would be able to support event-driven servers. It is also designed with Android apps in mind, so it can run on a mobile client.

A drawback is that Kotlin itself does not currently have bindings. However, since it is interoperable with Java, it is still able to use TensorFlow.

6. Conclusion

No other programming language has all the functionality of TensorFlow that Python does; however, for this application there definitely are other suitable languages. Both Java and Kotlin are good alternatives, offering better performance than Python. While OCaml could have some performance advantages as well, the fact that it is a functional language, with very different syntax from what most are used to, outweighs its benefits when languages like Java and Kotlin also exist. Between Java and Kotlin, Kotlin seems like a better choice, as it is designed to be basically Java but better.

7. References

Java Wikipedia Article
https://en.wikipedia.org/wiki/Java_(programming_language)
Kotlin Language Documentation
https://kotlinlang.org/
TensorFlow Architecture, Bindings
https://www.tensorflow.org/guide/extend/architecture
TensorFlow Java Binding GitHub
https://github.com/tensorflow/tensorflow/tree/master/tensorflow/java

TensorFlow OCaml Binding GitHub
https://github.com/LaurentMazare/tensorflow-ocaml
OCaml Wikipedia Article
https://en.wikipedia.org/wiki/OCaml