Язык Mojo для разработчиков ИИ — быстрее, чем Python

15.06.2023

Мир технологий постоянно движется. Новичком в этом мире является язык программирования Mojo. Он призван повысить скорость выполнения проектов на базе языка Python, обеспечивая разработчикам скорость, аналогичную языку C. Python — один из лучших языков программирования. Это универсальный и простой в изучении язык программирования, который открывает новичкам путь к программированию/компьютерным наукам.

Более того, это отличный язык программирования для рук компетентных разработчиков, которые могут использовать его для создания сложных приложений. Однако одним из самых существенных недостатков Python является скорость его выполнения. И именно здесь на помощь приходит Mojo. В этой статье мы расскажем о Mojo и о том, как он связан с экосистемой Python. Давайте начнем.

Что такое Mojo?

Mojo — это высокоуровневый современный язык программирования. Он предлагает интуитивно понятный дизайн, чтобы помочь разработчикам быстро создавать приложения. Кроме того, он стремится преодолеть разрыв между производством и исследованиями, позволяя пользователям использовать функции метапрограммирования и системного программирования с синтаксисом и экосистемой Python. Он в значительной степени заимствует Rust и обеспечивает высокую скорость выполнения в экосистеме Python. Технически Mojo является суперсетом Python, который предоставляет доступ.

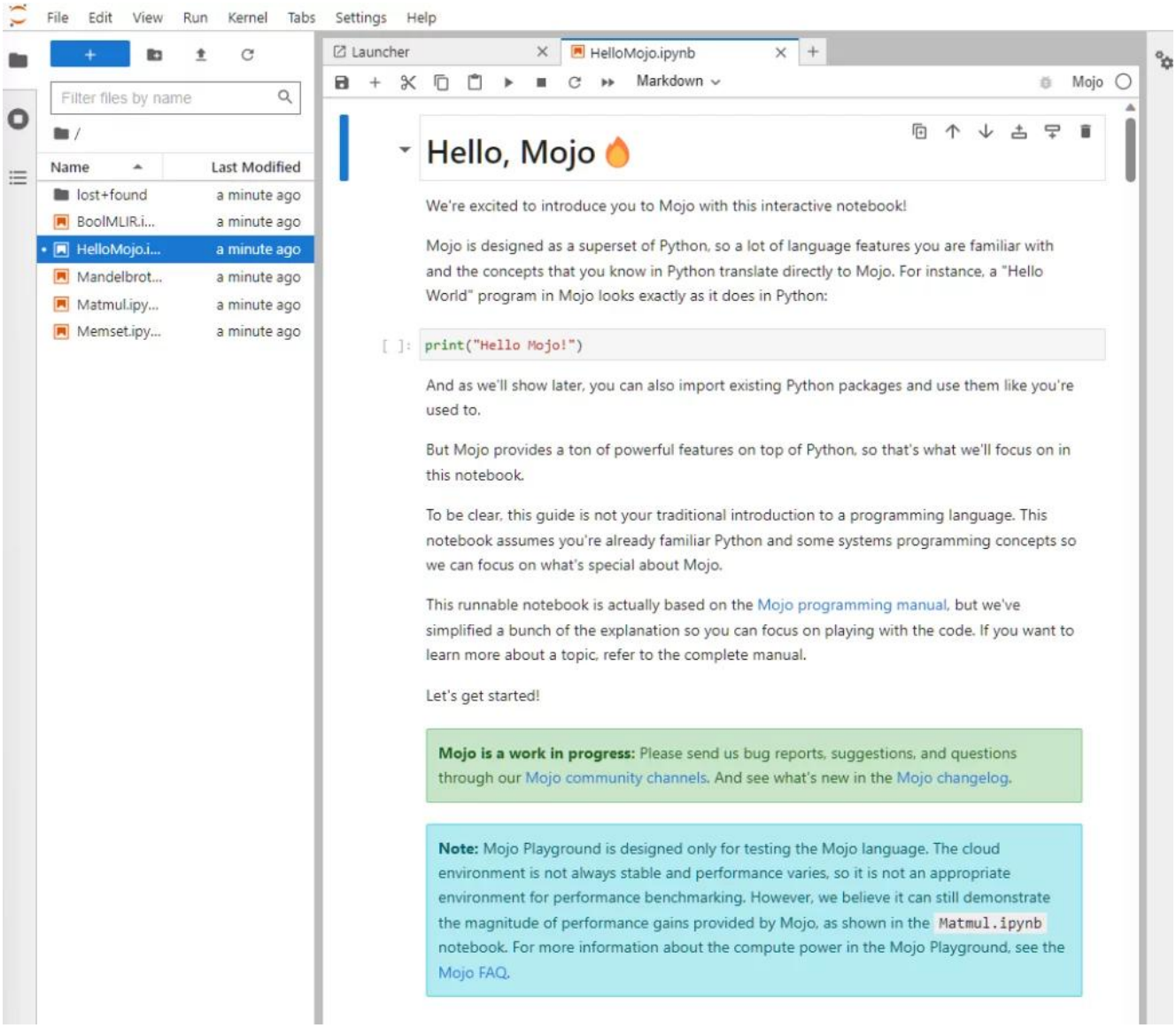
За Mojo стоит команда из Modular, компании, занимающейся инфраструктурой ИИ. И это также означает, что это язык программирования для разработчиков ИИ. Наряду с языком программирования Mojo, они также представили Interference Engine, который позволяет разработчикам улучшить рабочий процесс, масштабировать продукты ИИ и уменьшить задержку в выводах (подробнее об этом позже). По словам генерального директора Modular Криса Латтнера, Mojo в 35 000 раз быстрее, чем Python. Он также стоит за разработкой языка быстрого программирования. Увеличение скорости достигается благодаря тому, что Mojo использует инструментарий компилятора LLVM и инфраструктуру компилятора MILR (Multi-level Intermediate

Representation Overview).

Цели языков программирования Mojo включают:

- Работа с полной совместимостью с экосистемой Python.
- Предоставление разработчикам возможности развертывать подмножества кода на ускорителях.
- Низкоуровневый контроль для обеспечения предсказуемой производительности.
- Обеспечить отсутствие фрагментации экосистемы.

Чтобы попробовать Mojo, вам нужно использовать его через их облачную хостинговую среду Mojo Playground. Вам будет предложено войти в систему, и вы сможете получить рабочую среду!



Зачем нам нужен Mojo?

Основная идея Mojo заключается в унификации инфраструктуры ML/AI путем предоставления языка программирования, который работает во всем стеке. Кроме того, он обеспечивает простоту использования, устраняя необходимость написания MLIR-кода. Согласно Modular, Mojo предложит масштабируемую и инновационную модель программирования. Благодаря этому пользователям в области ИИ будет легко работать с ускорителями и гетерогенными системами. Технически Mojo является языком программирования, который поддерживает мета-программирование во время компиляции. Он также поддерживает другие возможности, такие как кэширование во время потока компиляции, адаптивные методы компиляции и т.д. Эти возможности отсутствуют в других

языках программирования.

Особенности языка программирования Mojo

В этом разделе мы рассмотрим ключевые особенности языка программирования Mojo.

Полная совместимость с Python

Mojo нацелен на работу с экосистемой Python, а не против нее. Это видно из того факта, что Mojo использует те же функции, библиотеки и возможности, которые предлагает Python. Таким образом, вы можете использовать любую библиотеку Python в Mojo.

Для импорта вам нужно использовать следующий код:

```
from PythonInterface import Python
```

После этого вы можете использовать Python.import_module() для импорта любой библиотеки Python. Например, чтобы импортировать numpy, вам нужно использовать следующую строку кода.

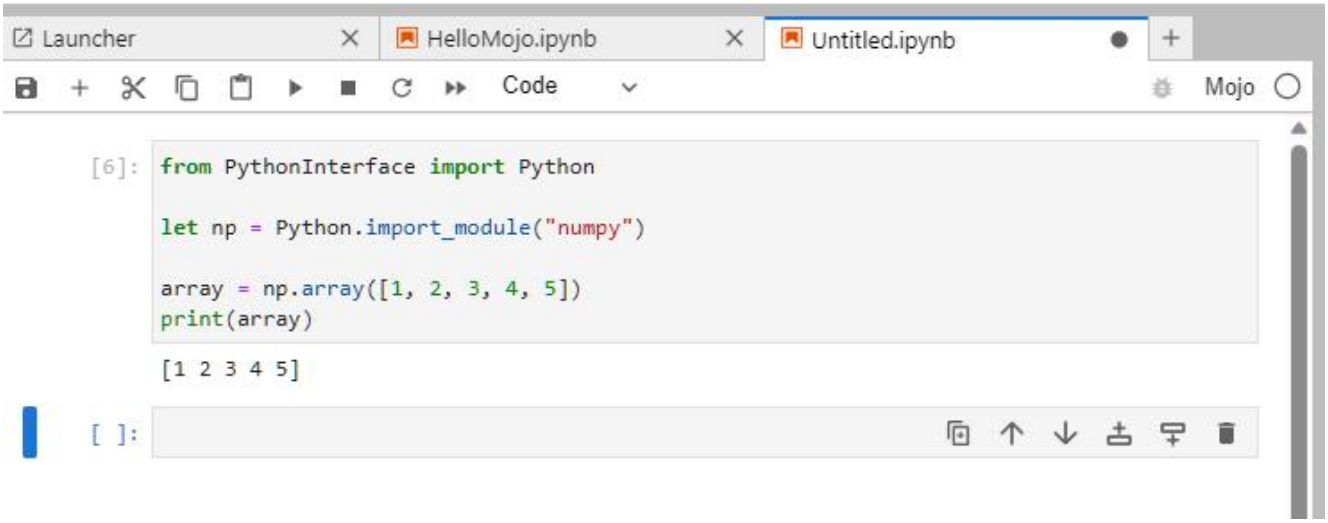
```
let np = Python.import_module(“numpy”)
```

В Python вам нужно выполнить команду “import numpy as np”. После импорта вы можете использовать его для создания массивов, выполнения вычислений и т.д.

```
array = np.array([1, 2, 3])

print(array)
```

Аналогично, вы можете импортировать matplotlib.pyplot для создания графика в Mojo. Вот как это выглядит, когда я запускаю код в Mojo Playground.



Если вы хотите опробовать код, скопируйте и вставьте его ниже.

```
from PythonInterface import Python

let np = Python.import_module("numpy")

array = np.array([1, 2, 3, 4, 5])

print(array)
```

MILR

MILR означает многоуровневое промежуточное представление. Mojo поддерживает MILR. Это, в свою очередь, позволяет разработчикам использовать полный набор новых расширенных возможностей. Эти возможности включают аппаратные блоки AI, потоки и векторы. MILR повышает производительность за счет параллелизма, делая Mojo быстрее, чем Python. Кроме того, он позволяет разработчикам использовать преимущества нескольких ядер.

Проверка владельца и заемщика

Управление памятью в Python безопасно. Он использует сборщик мусора, поэтому программисты должны убедиться, что код не столкнется с условиями гонки. Mojo, подобно Rust, реализует строгую модель проверки прав собственности и заемщиков. В настоящее время она реализована частично. Идея использования

этой модели заключается в улучшении параллелизма и обеспечении отличного управления памятью. Модель владения также обеспечивает потоко-безопасный подход, что идеально подходит для обеспечения превосходной поддержки параллелизма. Таким образом, программы не сталкиваются с условиями гонки. Кроме того, проверка заемщика гарантирует, что переменные всегда проверяются во время выполнения.

Нулевая стоимость абстракции

Mojo предлагает абстракции с нулевой стоимостью, позволяя разработчикам полностью контролировать хранение данных. Здесь программисты могут делать inline выделение значений структурам.

Автонастройка

Mojo также предлагает автонастройку. Это обеспечивает автоматическое выделение наилучших значений для параметров в зависимости от целевого оборудования. Автонастройка избавляет от необходимости вручную оптимизировать код в соответствии с целевым оборудованием.

Скорость: Насколько быстр язык Mojo?

Python – это высокоуровневый язык программирования, который стремится к простоте использования и удобству сопровождения. К сожалению, это делает его медленным по сравнению с другими решениями или языками программирования. В тесте Modular они обнаружили, что Mojo быстрее в 35000 раз. Они использовали алгоритм Мандельброта и запустили его на экземпляре AWS с процессором Intel Xeon. Они протестировали PYPY, SCALAR C++ и MOJO, а также Python. Результаты оказались очень быстрыми, и вы можете увидеть их ниже.

LANGUAGES	TIME (S) *	SPEEDUP VS PYTHON
PYTHON 3.10.9	1027 s	1x
PYPY	46.1 s	22x
SCALAR C++	0.20 s	5000x
MOJO 🔥	0.03 s	35000x

Чтобы узнать больше о скорости Mojo, ознакомьтесь с этим постом в сообществе Julialang.

Modular Inference Engine — дешевый запуск моделей ИИ

Modular также разрабатывает Modular Interference Engine, который позволяет удешевить запуск моделей ИИ в производстве. Mojo поддерживает Modular Interfence Engine по умолчанию. Он позволяет командам упростить рабочий процесс. Он также позволяет разработчикам сократить задержку в выводах, что упрощает масштабирование продуктов ИИ. Кроме того, разработчикам не нужно менять свою модель, чтобы использовать Engine. После загрузки он может повысить производительность моделей PyTorch и TensorFlow, способных работать с высокой производительностью при широкой аппаратной поддержке.

Заменит ли он Python?

Mojo — это новинка. Он выглядит многообещающим. Поэтому ему потребуется время, чтобы достичь своей целевой аудитории, такой как ученые, изучающие данные или языки программирования. И да, он решает конкретные проблемы для энтузиастов и изучающих ИИ. Однако существует множество аналогичных решений, которые повышают скорость работы с языком Python. Например, вы найдете Jax, Codon и Julia — язык, ориентированный на науку о данных. Таким образом, могут произойти две вещи. Во-первых, он

экспоненциально растет в плане возможностей, и сообщество принимает его. Другой исход — он станет специальным языком программирования, использующим библиотеки Python и механизм модульного вмешательства. Итак, заменит ли Mojo Python? Только время покажет.