Landscape of High-Performance Python to Develop Data Science and Machine Learning Applications

Georgina Sharon Aquino Nolasco

Introduction

- **Popularity of Python**: Python is one of the most widely used programming languages today, ranking first on the PYPL and TIOBE indices in 2022. Its success in data science (DS) and machine learning (ML) is largely due to its specialized libraries such as NumPy, Pandas, TensorFlow, Scikit-learn, SciPy, and Matplotlib.
- **Performance Limitations**: Python is relatively slow due to its interpreted nature and dynamic typing system. The standard implementation, CPython, uses the Global Interpreter Lock (GIL), which limits multi-threaded execution and affects performance in CPU-intensive operations.

Objective

- **Purpose**: Provide an organized overview of high-performance tools and techniques for Python in the DS and ML domains. Help practitioners find and use tools that improve the computational efficiency of their algorithms.
- **User Profile-Based Approach**: Identify suitable solutions according to different scenarios and needs faced by DS and ML practitioners.

Method and Approach

- **Narrative Review**: Qualitative evaluation of the various existing approaches in the field of Python performance improvement. This approach allows for a deep and contextualized understanding of the available tools.
- **Sources of Information**:Use of Google Scholar, GitHub, PyPI, Reddit, Stack Overflow, and other relevant sources to gather information on tools and techniques.
- **Inclusion and Exclusion Criteria**:Inclusion of tools that directly improve Python performance for DS and ML tasks. Exclusion of tools outside the Python domain and those specific to non-conventional hardware such as specialized devices other than CPU and GPU.

Categories of Tools and Techniques

- **Pure Python Performance Improvements**:
  Cython: Transforms Python code into C to enhance speed.
  Numba:Compiles Python functions to machine code using LLVM.
- **Parallelization and Distributed Execution**:
  Dask: Enables task parallelization and workload distribution across multiple cores and machines.
  Apache Spark: Framework for distributed processing of large datasets.
- **Specialized Libraries and Frameworks**:
  TensorFlow: Open-source library for numerical computation and deep learning.
  PyTorch: Flexible and efficient framework for deep learning, with GPU support.