Print to PDF

MLX

MLX is a NumPy-like array framework designed for efficient and flexible machine learning on Apple silicon, brought to you by Apple machine learning research.

The Python API closely follows NumPy with a few exceptions. MLX also has a fully featured C++ API which closely follows the Python API.

The main differences between MLX and NumPy are:

> - **Composable function transformations**: MLX has composable function transformations for automatic differentiation, automatic vectorization, and computation graph optimization.
> - **Lazy computation**: Computations in MLX are lazy. Arrays are only materialized when needed.
> - **Multi-device**: Operations can run on any of the supported devices (CPU, GPU, ...)

The design of MLX is inspired by frameworks like PyTorch, Jax, and ArrayFire. A notable difference from these frameworks and MLX is the *unified memory model*. Arrays in MLX live in shared memory. Operations on MLX arrays can be performed on any of the supported device types without performing data copies. Currently supported device types are the CPU and GPU.

Install

Build and Install

Usage

Quick Start Guide

Lazy Evaluation

Unified Memory

Indexing Arrays

Saving and Loading Arrays

Function Transforms

Compilation

Examples

Python API Reference

C++ API Reference

Further Reading