



2024 年度 未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当 PM

曾川 景介（newmo 株式会社 CTO）

2. クリエータ氏名

道本 将弘（埼玉大学 工学部 電気電子物理工学科）

3. 委託金支払額

2,880,000 円

4. テーマ名

型安全でクロスプラットフォームな次世代 ML ライブラリの開発

5. 関連 Web サイト

Rusty Lantern ライブラリの公式広報 X アカウント：

https://x.com/RustyLantern_rs

6. テーマ概要

本プロジェクトは、型安全かつクロスプラットフォーム対応の次世代機械学習ライブラリ「Rusty Lantern」を開発した。昨今の AI や ML 界限では一部のベンダによる寡占がかなり顕著になっている。多くのライブラリも著しいベンダベンディングが行われて特定のベンダに依存する実装が一般的になっている。そのような状況のなかで、複数の問題を同時に解決してしまうような大胆な次世代 ML ライブラリを提案した。現在のニューラルネットワーク開発における環境依存やランタイムエラーの課題を解決するため、Rust 言語を用いてゼロから設計し、計算バックエンドとして WebGPU を活用することで高い可搬性を実現している。また CPU も計算バックエンドとして選択できるように設計されている。

加えて、Rust の型システムを活かしたコンパイル時の型検査により、ランタイムエラーを抑制し、開発の効率化を図っている。さらに、GUI デバッガ「LanternBoard」による可視化機能を備え、開発者のデバッグ負担を軽減するなど ML ライブラリの開発に必要な周辺機能も合わせて提供している。

7. 採択理由

どのような OS やどのベンダの GPU であっても動作することを目指した、まさに次世代の ML ライブラリを開発しようとする提案である。加えて提案者は、型システムを導入することでデバッグの容易性や開発効率の向上も見込めると考えている。本提案のユニークなところは GPU の抽象化レイヤーとして WebGPU を用いることである。現在主流となっている ML や AI の環境は特定のベンダのプロプライエタリなライブラリに大きく依存しており、今後のこの領域の発展には本提案が貢献できる余地があると評価した。また提案者は、一つの問題を解決するだけでなく、他の問題も同時に解決しようという大胆な発想を持っており、OS や GPU ベンダに対してクロスプラットフォームであるだけでなく、同時に型安全なライブラリとして提供しようとする点も評価した。

8. 開発目標

本プロジェクトの開発目標は、現代のニューラルネットワーク開発における環境依存やランタイムエラーの多発といった根本的な課題を解決するために、従来のライブラリやエコシステムとは異なる新しいアーキテクチャを提示し、その原型となる機械学習ライブラリとエコシステムを構築することにある。特に、既存ライブラリに見られる Python が動的型付けのスクリプト言語であることによる非効率性や、GPU 環境の違いによるコードの移植性の低さといった問題を克服するために、Rust 言語と WebGPU を組み合わせることで、高い型安全性とクロスプラットフォーム対応を両立する仕組みの実現を目指した。

ライブラリの設計においては、行列の形状や演算の互換性をコンパイル時に検証できる独自の型システムを導入し、デバッグコストの削減とコードの堅牢性向上を図ることが目標の一つである。これによって、同一コードを PC、モバイル、クラウドといった多様な環境で変更なく実行可能にすることで、開発環境構築の手間を軽減し、最終的には再現性の高い開発基盤を提供することを目指した。

なお、開発の過程で得られた知見に基づき、当初の計画には含まれていなかった GUI デバッグの導入も実現された。これは、内部状態の可視化による開発支援を目的としたものであり、実装中に見出されたニーズに応じて機能拡張が行われた。

9. 進捗概要

本プロジェクトにおける開発は、当初掲げた目標に対して着実かつ高水準で進行された。Rust 言語を用い、外部の数値計算ライブラリに依存することなく四則演算レベルから実装された Rusty Lantern 本体には、CPU および WebGPU による計算バックエンドが搭載されており、クロスプラットフォーム対応と高

い可搬性を実現する基盤が確立された。

加えて、行列演算や操作を担う LanternCore、データ前処理を行う LanternDataset、計算グラフと自動微分を提供する LanternNN といった複数のモジュールが一貫した設計思想のもとで構築されており、それぞれが高い抽象性と型安全性を備えている。これにより、実装ミスの抑制やデバッグ効率の向上といった実用性の高い開発体験を提供することが可能となっている。

さらに、開発の過程で見出されたニーズに応じて、当初の計画には含まれていなかった GUI デバッガ「LanternBoard」が追加で実装された。これにより、実行時の内部状態を視覚的に確認しながら操作できるようになり、開発中の問題解決を一層効率化する機能が実現された。

一方で、本ライブラリの実用性と性能を検証するために、既存のライブラリで構築された言語モデルを Rusty Lantern 上で再実装し、比較評価を行う計画も進められていたが、成果報告時点ではその実装および比較作業が完了には至らなかった。今後、既存ライブラリとの比較を通じて本ライブラリの優位性を定量的に示すことを期待している。

10. プロジェクト評価

本プロジェクトは、既存の Python ベースの機械学習ライブラリが抱える環境依存やランタイムエラーの多発といった課題を、Rust を活用することで根本的に解決した、独自の型システムを導入し、コンパイル時に計算互換性を検証することで、実装の堅牢性が高めることに成功した。さらに、WebGPU を活用したクロスプラットフォーム対応により、環境構築の負担を軽減する点も実用的で優れている。

機械学習ライブラリ分野では、PyTorch や TensorFlow がデファクトスタンダードとなっているが、Rust をベースにした型安全なライブラリを実現することは野心的な挑戦であった。クロスプラットフォームを実現するために ML ライブラリのバックエンドに WebGPU を活用している点も革新的であり、将来的にブラウザ上での実行も可能になる新たな応用可能性も示唆している。

しかし、現状のデファクトスタンダードと比較した場合の優位性を本プロジェクト期間内に十分に実証できなかったことが残念であった。

11. 今後の課題

プロジェクト期間内にやりきれなかった既存ライブラリとの優位性の比較が挙げられる。既存ライブラリ上で動いている言語モデルを本プロジェクトで作成したライブラリで動かすことで、優位性を比較してほしい。また本プロジェクトで作成したライブラリ自体も、OSS 化していくことや実際に開発者からフィードバックを得ることで、さらにブラッシュアップすることが必要である。