tracel-ai / **burn** Public

main

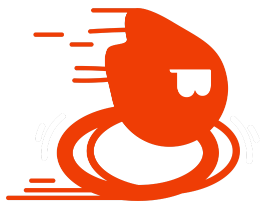
Go to file

unrenormalizable ✓ yesterday

chat 58 online | crates.io v0.11.1 | docs latest | test passing | codecov 85% | Rust 1.71.0+ | license MIT/Apache-2.0

Burn is a new comprehensive dynamic Deep Learning Framework built using Rust with extreme flexibility, compute efficiency and portability as its primary goals.

Performance



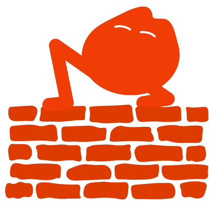
Because we believe the goal of a deep learning framework is to convert computation into useful intelligence, we have made performance a core pillar of Burn. We strive to achieve top efficiency by leveraging multiple optimization techniques described below.

Click on each section for more details 👇

- ▶ Automatic kernel fusion 💥
- ▶ Asynchronous execution ❤️‍🔥
- ▶ Thread-safe building blocks 🦞
- ▶ Intelligent memory management 🦀
- ▶ Automatic kernel selection 🎯
- ▶ Hardware specific features 🔥

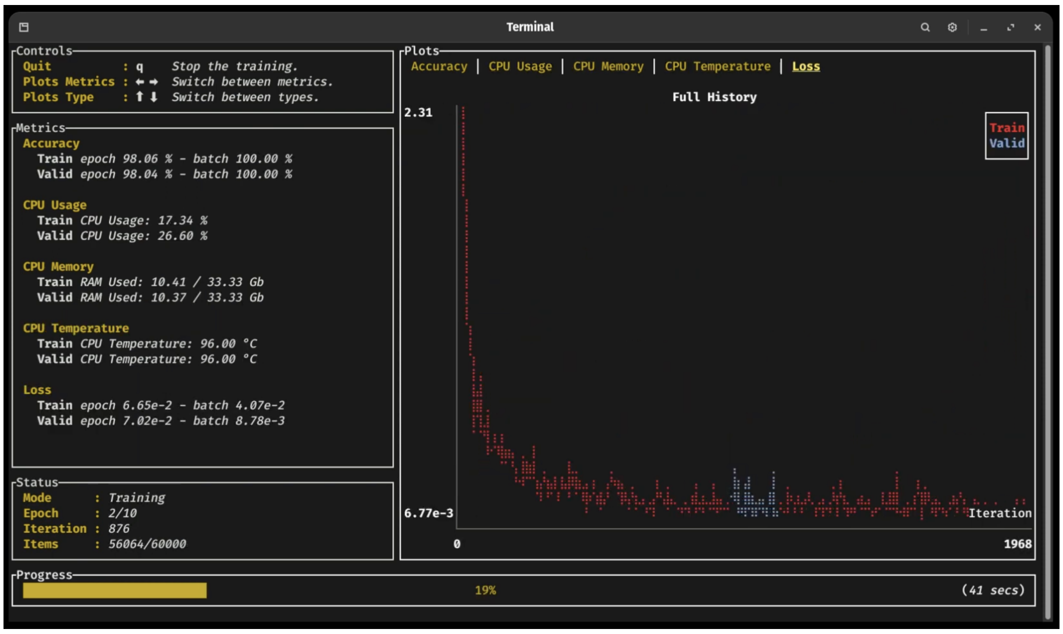
▶ Custom Backend Extension

Training & Inference



The whole deep learning workflow is made easy with Burn, as you can monitor your training progress with an ergonomic dashboard, and run inference everywhere from embedded devices to large GPU clusters.

Burn was built from the ground up with training and inference in mind. It's also worth noting how Burn, in comparison to frameworks like PyTorch, simplifies the transition from training to deployment, eliminating the need for code changes.



```
┌Controls──────────────────────────────────────────────
 Quit          : q     Stop the training.
 Plots Metrics : ← →   Switch between metrics.
 Plots Type    : ↑ ↓   Switch between types.

┌Metrics───────────────────────────────────────────────
 Accuracy
   Train epoch 98.06 % - batch 100.00 %
   Valid epoch 98.04 % - batch 100.00 %

 CPU Usage
   Train CPU Usage: 17.34 %
   Valid CPU Usage: 26.60 %

 CPU Memory
   Train RAM Used: 10.41 / 33.33 Gb
   Valid RAM Used: 10.37 / 33.33 Gb

 CPU Temperature
   Train CPU Temperature: 96.00 °C
   Valid CPU Temperature: 96.00 °C

 Loss
   Train epoch 6.65e-2 - batch 4.07e-2
   Valid epoch 7.02e-2 - batch 8.78e-3

┌Status────────────────────────────────────────────────
 Mode      : Training
 Epoch     : 2/10
 Iteration : 876
 Items     : 56064/60000

┌Plots─────────────────────────────────────────────────
 Accuracy | CPU Usage | CPU Memory | CPU Temperature | Loss

                      Full History
 2.31                                              Train
                                                   Valid
 6.77e-3                                       Iteration
        0                                           1968

┌Progress──────────────────────────────────────────────
                     19%                      (41 secs)
```

Click on the following sections to expand

▶ Training Dashboard

▶ ONNX Support

▶ Inference in the Browser

▶ Embedded: *no_std* support

Backends

Burn strives to be as fast as possible on as many hardwares as possible, with robust implementations. We believe this flexibility is crucial for modern needs where you may train your models in the cloud, then deploy on customer hardwares, which vary from user to user.

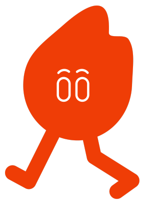
Compared to other frameworks, Burn has a very different approach to supporting many backends. By design, most code is generic over the Backend trait, which allows us to build Burn with swappable backends. This makes composing backend possible, augmenting them with additional functionalities such as autodifferentiation and automatic kernel fusion.

We already have many backends implemented, all listed below 👇

- WGPU (WebGPU): Cross-Platform GPU Backend 🌐
- Candle: Backend using the Candle bindings 🕯
- LibTorch: Backend using the LibTorch bindings 🎆
- NdArray: Backend using the NdArray primitive as data structure 🦐
- Autodiff: Backend decorator that brings backpropagation to any backend 🔄
- Fusion: Backend decorator that brings kernel fusion to backends that support it 💥

Getting Started

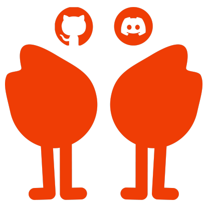
Just heard of Burn? You are at the right place! Just continue reading this section and we hope you can get on board really quickly.



- The Burn Book 🔥
- Examples 🙏
- Pre-trained Models 🤖
- Why use Rust for Deep Learning? 🦀

Community

If you are excited about the project, don't hesitate to join our [Discord](https://discord.gg)! We try to be as welcoming as possible to everybody from any background. You can ask your questions and share what you built with the community!



Contributing

Before contributing, please take a moment to review our code of conduct. It's also highly recommended to read our architecture document, which explains some of our architectural decisions. Refer to out contributing guide for more details.

Status

Burn is currently in active development, and there will be breaking changes. While any resulting issues are likely to be easy to fix, there are no guarantees at this stage.

License

Burn is distributed under the terms of both the MIT license and the Apache License (Version 2.0). See LICENSE-APACHE and LICENSE-MIT for details. Opening a pull request is assumed to signal agreement with these licensing terms.