From TensorFlow to PyTorch

With some help from Rust

Gavrie Philipson
Rusty Bits Software Ltd.
June 2025

About Me

Gavrie Philipson

- Rust, Python, Cloud, Backend, DevOps, and more.
- Bootstrapping software development teams: Training, mentoring, and hiring
- Consulting to startup companies on software development and architecture

Rusty Bits Software Ltd.
https://rustybits.io
gavrie@rustybits.io

About You

Using Rust to improve Python

Astral

PyO_3

The Mission

- Port ML model from TensorFlow to PyTorch
- Lots of training data in `TFRecord` format

The `TFRecord` format

- A sequence of `HashMap<String, Vec<T>>`
- `where T: u8 | f32 | i64`
- Serialized with `protobuf`

`TFRecord` Example

```
[
  {
    "label": "cat",
    "image/shape": [320, 200, 3],
    "image/encoded": [0x12, 0x34, 0x56, ...],
  },
  {
    "label": "dog",
    "image/shape": [320, 200, 3],
    "image/encoded": [0x78, 0x9a, 0xbc, ...],
  },
]
```

The Constraints

- Dependencies (look at venv size)
- Performance: Keep GPUs busy
- Ease of use for Python devs

Challenge: Getting Test Data

- No access to the original data
- Vibe code some Python to generate test data!

Playing on Rust's strengths

- Designing with types
- Dive into the Rust implementation and `tests.rs`

The End Result

- `pip install rustfrecord`
- `test_rustfrecord.py`
- `src/lib.rs`

Getting the Code

https://pypi.org/project/rustfrecord/

https://github.com/gavrie/rustfrecord