Candle 🕯️

A Minimalist ML Framework

Desert Rust 2025-07-09

What is an ML Framework?

- Library for building machine learning models
  - Model structure
  - Weight serialization
- Training
  - Optimizers
  - Data loaders
  - Data randomizers
  - Backpropagation
- Inference

A Brief History...

- Early ML frameworks were written in Python and C++
- Initially there were a lot of home grown frameworks and little standardization
- Consolidation around TensorFlow(Google) and PyTorch(Facebook)
- PyTorch takes the lionshare of ML development
    - Largely due to its simplified execution model

Python ML Frameworks

Pros:

- Easy to get started
- Support for the latest models
- Most common used language in ML

Cons:

- Python
- Heavyweight
- Slow

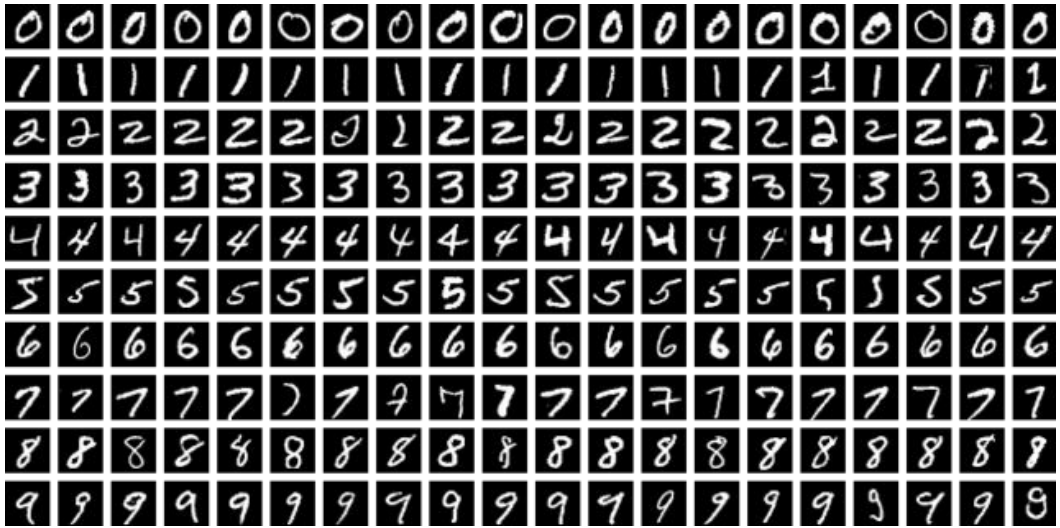
Okay...what is Candle though?

🕯️ = tiny 🔥

What is Candle?

- Simple
- Lightweight
- Fast
- Familiar
- Rust

Simple



```rust
use candle_core::{Device, Result, Tensor};
use candle_nn::{Linear, Module};

struct Model {
    first: Linear,
    second: Linear,
}

impl Model {
    fn forward(&self, image: &Tensor) -> Result<Tensor> {
        let x = self.first.forward(image)?;
        let x = x.relu()?;
        self.second.forward(&x)
    }
}

fn main() -> Result<()> {
    // Use Device::new_cuda(0)?; to use the GPU.
    let device = Device::Cpu;

    // This has changed (784, 100) -> (100, 784) !
    let weight = Tensor::randn(0f32, 1.0, (100, 784), &device)?;
    let bias = Tensor::randn(0f32, 1.0, (100, ), &device)?;
    let first = Linear::new(weight, Some(bias));
    let weight = Tensor::randn(0f32, 1.0, (10, 100), &device)?;
    let bias = Tensor::randn(0f32, 1.0, (10, ), &device)?;
    let second = Linear::new(weight, Some(bias));
    let model = Model { first, second };

    let dummy_image = Tensor::randn(0f32, 1.0, (1, 784), &device)?;

    let digit = model.forward(&dummy_image)?;
    println!("Digit {digit:?} digit");
    Ok(())
}
```

A Lightweight Framework

Framework	Peak RAM	Memory Growth
Candle	3.2 GB	18 MB/min
torch-rs	4.7 GB	42 MB/min

Data from https://markaicode.com/rust-ai-frameworks-candle-pytorch-comparison-2025/

A Fast Framework

Framework	BERT Base (ms)	ResNet-50 (ms)	LLaMA 2 7B (ms/token)
Candle	8.3	12.6	45.2
torch-rs	15.7	19.4	72.8

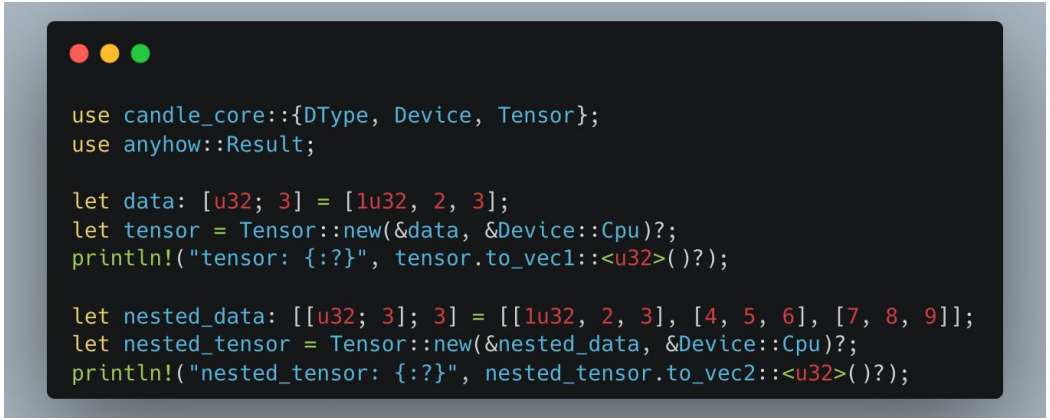
Data from https://markaicode.com/rust-ai-frameworks-candle-pytorch-comparison-2025/

Familiar

```python
import torch
from typing import List

data: List = [1, 2, 3]
tensor = torch.tensor(data)
print(tensor)

nested_data = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
nested_tensor = torch.tensor(nested_data)
print(nested_tensor)
```



```rust
use candle_core::{DType, Device, Tensor};
use anyhow::Result;

let data: [u32; 3] = [1u32, 2, 3];
let tensor = Tensor::new(&data, &Device::Cpu)?;
println!("tensor: {:?}", tensor.to_vec1::<u32>()?);

let nested_data: [[u32; 3]; 3] = [[1u32, 2, 3], [4, 5, 6], [7, 8, 9]];
let nested_tensor = Tensor::new(&nested_data, &Device::Cpu)?;
println!("nested_tensor: {:?}", nested_tensor.to_vec2::<u32>()?);
```

Familiar

```python
zero_tensor = torch.zeros_like(tensor)
ones_tensor = torch.ones_like(tensor)
random_tensor = torch.rand_like(tensor)
```

```rust
let data: [u32; 3] = [1u32, 2, 3];
let tensor = Tensor::new(&data, &Device::Cpu)?;

let zero_tensor = tensor.zeros_like()?;
println!("zero_tensor: {:?}", zero_tensor.to_vec1::<u32>()?);

let ones_tensor = tensor.ones_like()?;
println!("ones_tensor: {:?}", ones_tensor.to_vec1::<u32>()?);

let random_tensor = tensor.rand_like(0.0, 1.0)?;
println!("random_tensor: {:?}", random_tensor.to_vec1::<f64>()?);
```

Familiar

	Using PyTorch	Using Candle
Creation	`torch.Tensor([[1, 2], [3, 4]])`	`Tensor::new(&[[1f32, 2.], [3., 4.]], &Device::Cpu)?`
Creation	`torch.zeros((2, 2))`	`Tensor::zeros((2, 2), DType::F32, &Device::Cpu)?`
Indexing	`tensor[:, :4]`	`tensor.i((.., ..4))?`
Operations	`tensor.view((2, 2))`	`tensor.reshape((2, 2))?`
Operations	`a.matmul(b)`	`a.matmul(&b)?`
Arithmetic	`a + b`	`&a + &b`
Device	`tensor.to(device="cuda")`	`tensor.to_device(&Device::new_cuda(0)?)?`
Dtype	`tensor.to(dtype=torch.float16)`	`tensor.to_dtype(&DType::F16)?`
Saving	`torch.save({"A": A}, "model.bin")`	`candle::safetensors::save(&HashMap::from([("A", A)]), "model.safetensors")?`
Loading	`weights = torch.load("model.bin")`	`candle::safetensors::load("model.safetensors", &device)`

Rust

- Typesafe*
- Expressiveness
- Tooling
- Performance
- WASM

Demos!! 

Demo Links

- https://huggingface.co/spaces/lmz/candle-yolo
- https://huggingface.co/spaces/lmz/candle-whisper
- https://huggingface.co/spaces/lmz/candle-llama2
- https://huggingface.co/spaces/radames/candle-segment-anything-wasm
- https://huggingface.co/spaces/radames/Candle-BLIP-Image-Captioning