White Box Neural Network

Vishal Kalathil

October 21, 2022

1 Introduction

In this report I will go over my experiences with implementing a whitebox ANN in Rust. For this task I decided to create a simple neural network that would attempt to recognize the digits from the MNIST dataset.

2 Usage

Run

```
cargo run Cargo.toml
```

after installing rust and cargo

3 Implementation

The ANN reads data from a modified version of the MNIST dataset that was converted into a file for easier reading. I also trained the model on the test dataset (which consisted of 10000 entries and tested the performance using the first 200 elements in the train data set, as I was running short of computational resources)

The ANN also has:

- 28^2 nodes in the input layer
- 10 nodes in the hidden layer
- 10 nodes in the output layer

I also ended up using 2 different activation function combinations to see which one performed better. The first was the sigmoid function and the other was a combination of the ReLU function and the Softmax function.

4 Performance

At 1000 epochs with learning rate = 0.2

Table 1: Accuracy metrics

Activation Function	**Accuracy on Train Data**	**Accuracy on Test Data**
Sigmoid	0.9213	0.915
ReLU + Softmax	abcdef ghjijklmn	123.456778

5 Final Thoughts

Despite the many, many places this project can be improved upon like

1. Multithreading
2. Cleaner and more optimized code
3. Fix all the warnings
4. Build to a crate (?)
5. Use the actual dataset
6. Use GPU throught the RustCUDA project

The neural networks do perform decently well.