## **Optimizing MNIST Classification Using CUDA**

The MNIST dataset, consisting of 28×28 grayscale images of handwritten digits (0–9), is a widely used benchmark for evaluating image classification algorithms. The goal of this project is to implement and optimize a neural network-based solution for classifying MNIST digits, with a focus on leveraging NVIDIA GPUs through CUDA to significantly accelerate training and inference. The optimization process is approached incrementally, with each version introducing performance improvements or architectural enhancements. The starting point is a basic sequential CPU implementation, which serves as a reference for evaluating the gains achieved in subsequent CUDA-based versions.

### **V1 Sequential Implementation**

The initial version (V1) of the neural network for MNIST classification is a straightforward sequential implementation written in C. It features a single hidden-layer architecture with 128 neurons using ReLU activation and a softmax-activated output layer for multi-class classification. All computations, including matrix operations and gradient updates, are performed using nested loops without any parallelism or hardware acceleration. Data is loaded directly from the MNIST binary files, with pixel values normalized and labels one-hot encoded. Training is conducted using stochastic gradient descent over individual samples, and accuracy and loss are reported per epoch. While functional and clear, this version prioritizes simplicity over performance and does not yet utilize batching, vectorization, or optimized numerical libraries.