

Challenge #16: Benchmarking SAXPY with PyTorch

Objective:

Compare the performance of a simple feedforward neural network implemented using CUDA vs PyTorch.

Summary of Implementations:

CUDA Version:

- 4 inputs -> 5 hidden -> 1 output neuron.
- Uses custom CUDA kernels for forward propagation.
- Includes sigmoid activation and bias terms.
- Handles memory allocation and synchronization properly.
- Lacks performance timing and batch input support.

PyTorch Version:

- Same architecture (4-5-1) using torch.nn.Linear layers.
- Sigmoid activation applied in forward pass.
- Random weight initialization matches CUDA's method.
- Runs on GPU if available, using inference mode.
- No benchmarking or scalability test included.

Next Steps:

1. Add execution time measurement to both implementations:
 - CUDA: use cudaEventRecord for timing.
 - PyTorch: use Python's time.time().

- 2. Test with increased network size (e.g., more neurons/layers).
- 3. Run multiple iterations and average timing results.
- 4. Compare performance and analyze which approach is faster and under what conditions.

Conclusion:

Both implementations are functionally correct. To complete the challenge, performance benchmarking and scalability tests are needed.

Benchmark Comparison (Mock Data):

