# Project Proposal: Parallelized Matrix Multiplication and Extension to Neural Networks

Hritvik JV

October 27, 2025

## 1 Introduction and Motivation

Matrix multiplication is a foundational operation in scientific computing, graphics, and machine learning. However, it is also computationally intensive, especially for large matrices, with a time complexity of $O(n^3)$ for naïve implementations. This project aims to explore parallelization techniques to accelerate matrix multiplication, leveraging concepts learned in class to understand performance improvements across different architectures.

The motivation for choosing this area comes from its fundamental role in many modern computational systems—particularly deep learning and numerical simulations. Implementing an efficient parallelized matrix multiplier provides an excellent starting point for more complex parallel applications.

## 2 Problem Description

The initial goal is to implement a simple parallelized matrix multiplier using shared or distributed memory models (e.g., OpenMP, MPI, or CUDA). The objective is to analyze how computation time scales with matrix size and number of processing units, focusing on load balancing, synchronization, and data communication overhead.

Once this is achieved, the next step is to integrate the parallelized matrix multiplier into a basic multi-layer perceptron (MLP). The idea is to replace the standard matrix

multiplication in forward propagation with the custom parallelized version to demonstrate potential speedups in neural network computations.

## 3 Possible Directions of Investigation

Several interesting avenues of exploration include:

- Comparing different parallelization strategies: block vs. row-wise decomposition.
- Studying performance scaling on CPU vs. GPU implementations.
- Measuring communication overhead and synchronization costs.
- Evaluating accuracy and runtime trade-offs in the MLP when using different levels of precision (e.g., float vs. double).

## 4 Expected Results

The expected outcome is a working parallel matrix multiplication module that achieves measurable speedups over a serial implementation. Integration into a multi-layer perceptron should demonstrate how parallelization can enhance machine learning performance, even on modest hardware. The project will provide valuable insights into data partitioning, synchronization, and scalability—core principles of parallel computing.

## 5 Future Extensions and Applications

Beyond the initial implementation, this work could be extended into:

- Parallelized convolution operations for image processing.
- Parallel gradient computation for training larger models.
- Real-time matrix computations for simulation environments (e.g., physical simulations, graphics, or agent-based models).

## 6 Conclusion

This project will explore the practical aspects of parallel computing by implementing and testing a fundamental computational kernel: matrix multiplication. Extending this to an MLP will demonstrate how low-level parallel optimization connects to higher-level applications in artificial intelligence. The project serves as a bridge between theoretical concepts and their tangible impact on performance.