Neural network training

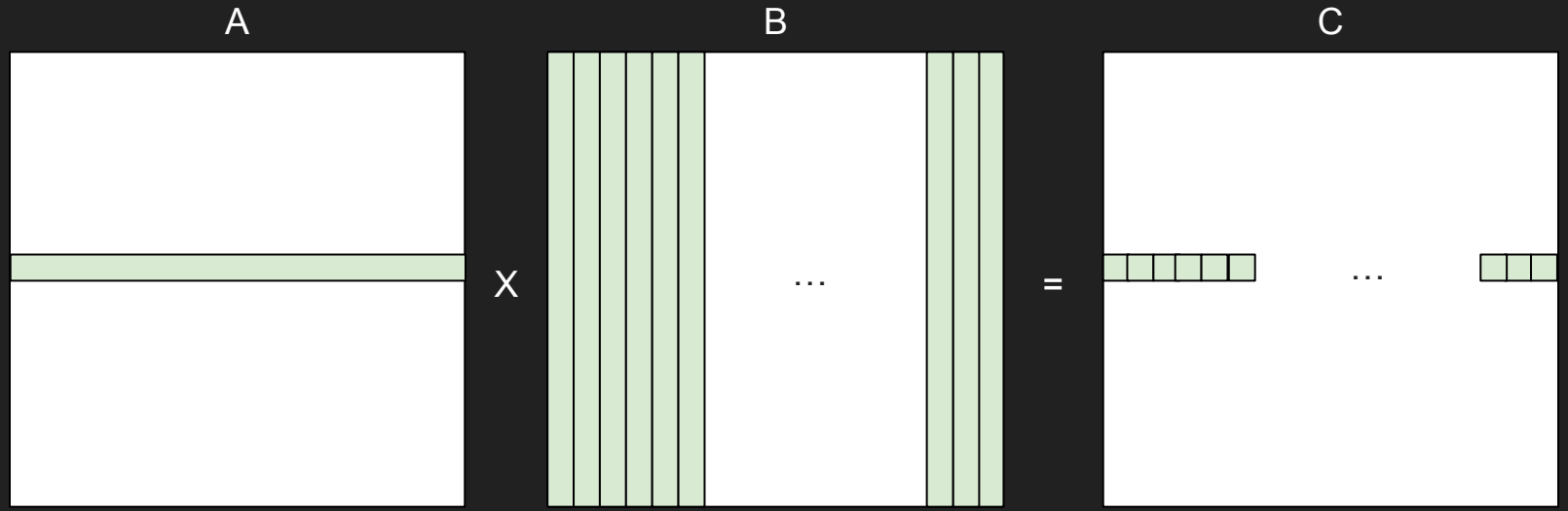
Gal Lindič, Ožbej Golob, Žan Jonke

Neural network training

- Forward pass algorithm
- Loss calculation
- Backward pass algorithm
- The network has 1 hidden of variable size layer
- Train set size is 24421 and test set size is 8140 with 14 features
- We experimented with varying batch size and hidden layer size

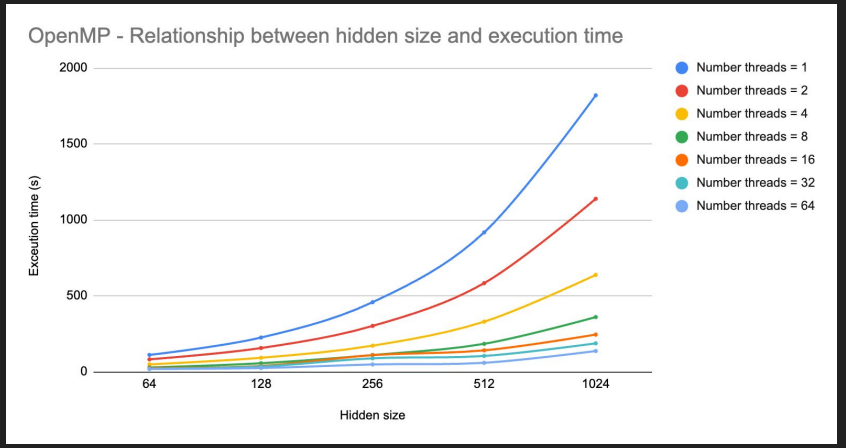
OpenMP implementation

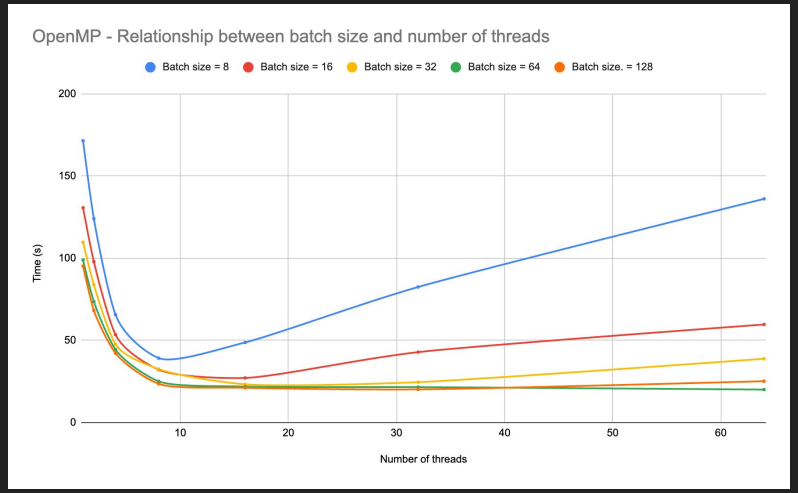
- Parallelization done row-wise ie. one thread per row
  - In matrix multiplication one thread per dot product
  - In layer activation one thread per row
  - In loss calculation one thread per sample
  - In hadamard product one thread per row

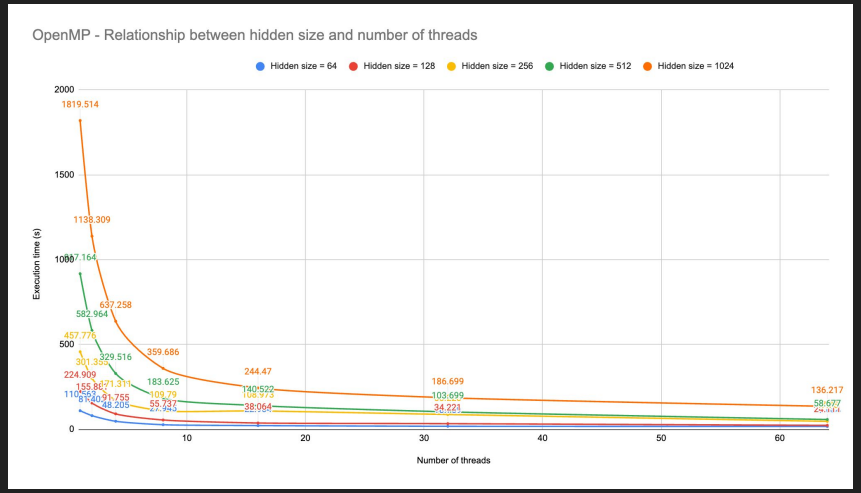


OpenMP benchmarking

- Increasing the number of threads pays off for larger hidden size
- For small batch sizes computing on large
- number of cores does not pay of

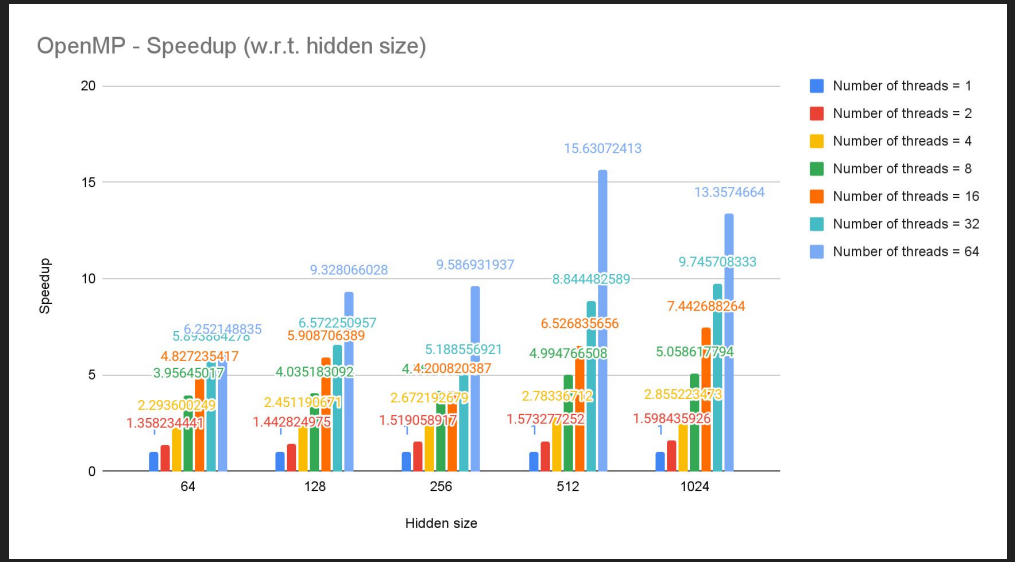


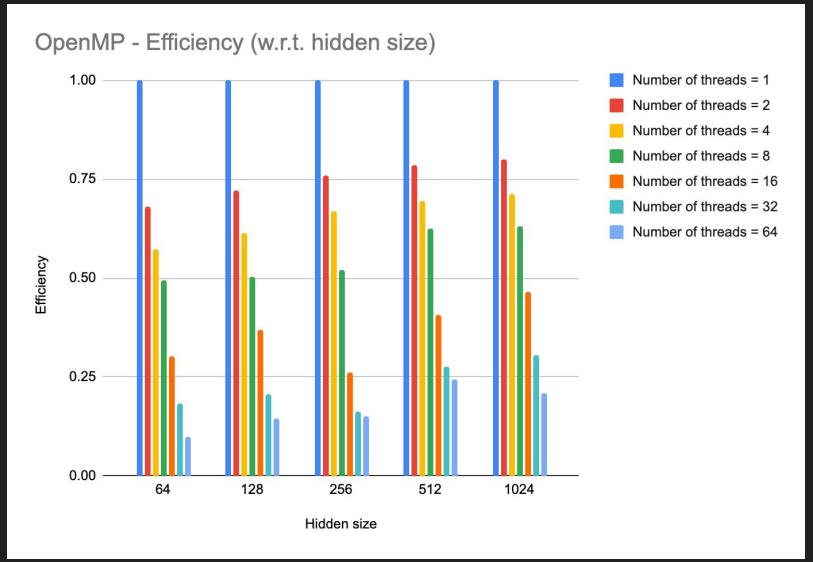




OpenMP benchmarking

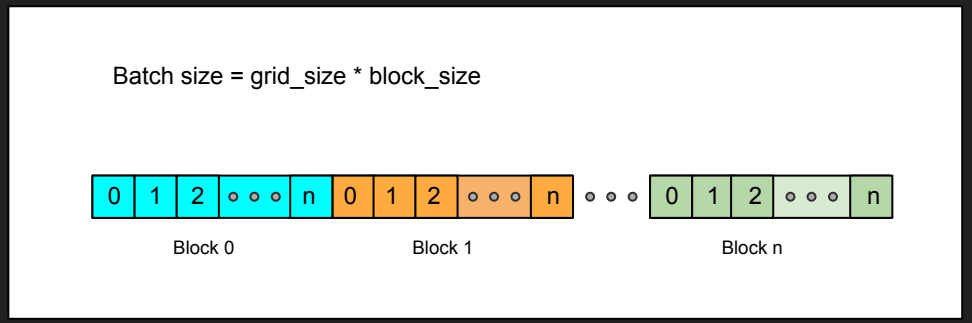
- Noticeable increase in speedup for larger hidden size
- Efficiency deteriorates





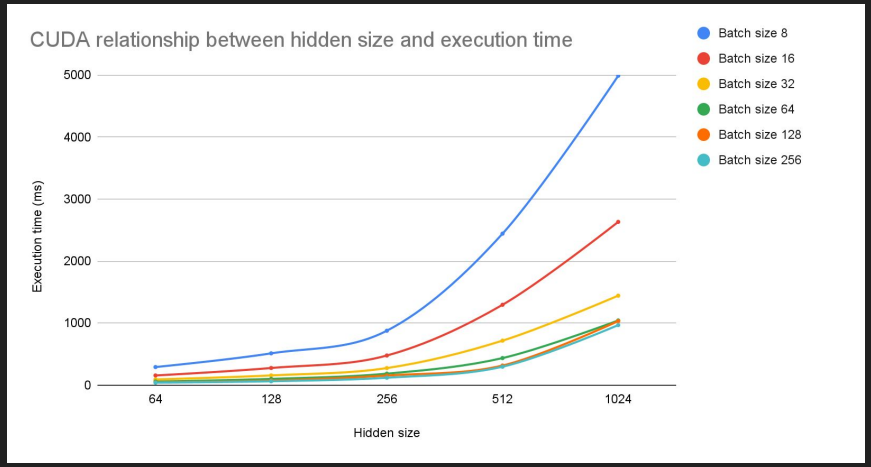
CUDA implementation

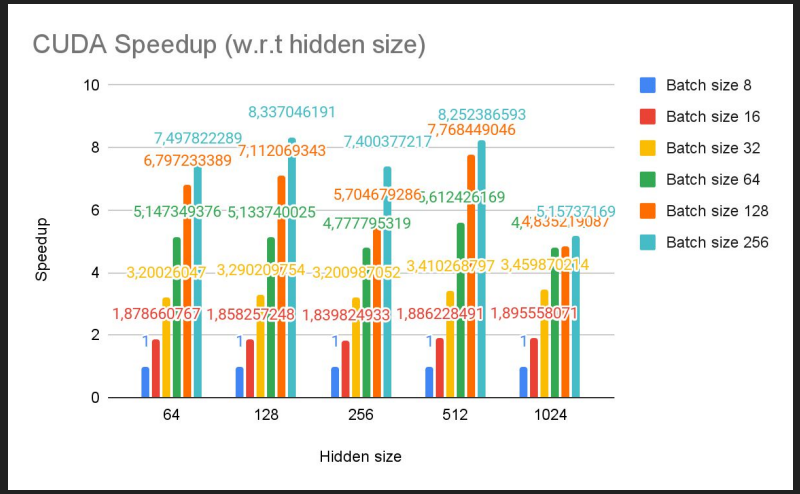
- Multiple kernels that handle different calculation tasks
- Dataset and MLP matrices stored in global GPU memory
- Number of threads in a kernel equals batch size
- Parallelization done row-wise
  - Each thread handles one row from batch data
  - Cases when there are matrices with more rows than threads, we divide the rows equally among threads

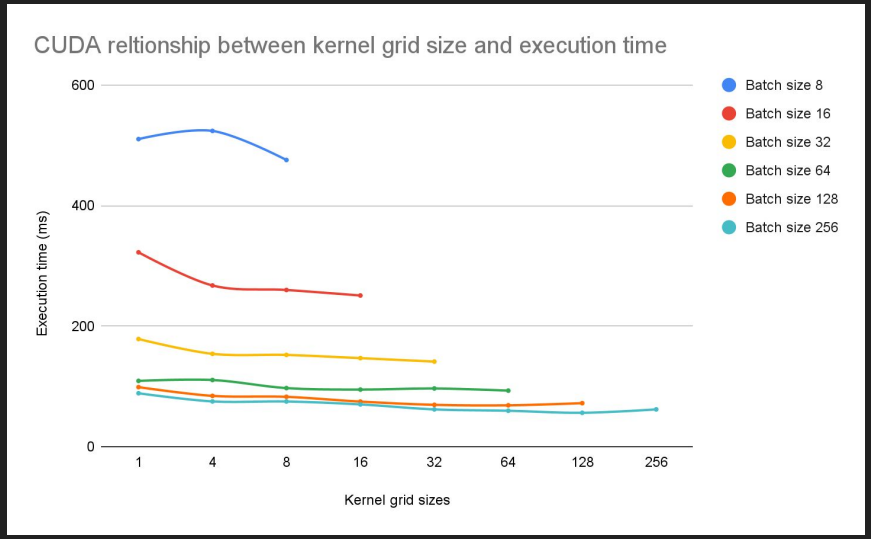


CUDA benchmarking

- Increase in batch size greatly improves execution time
- Having higher grid sizes (more blocks in kernel) slightly improves execution time

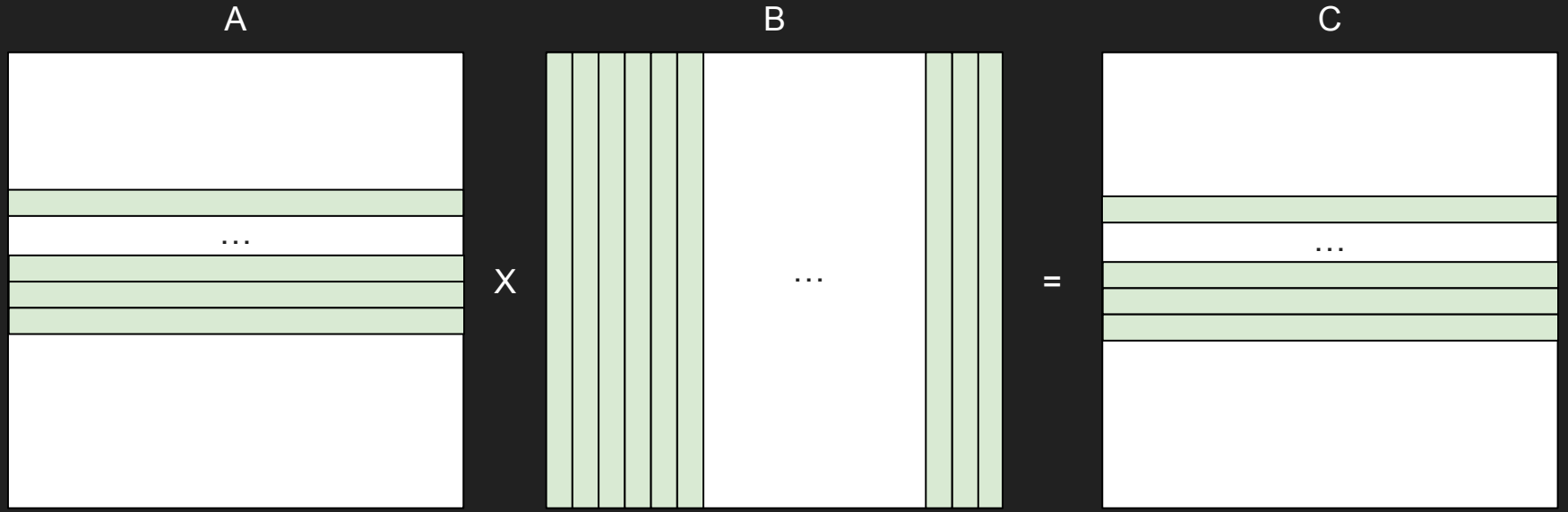




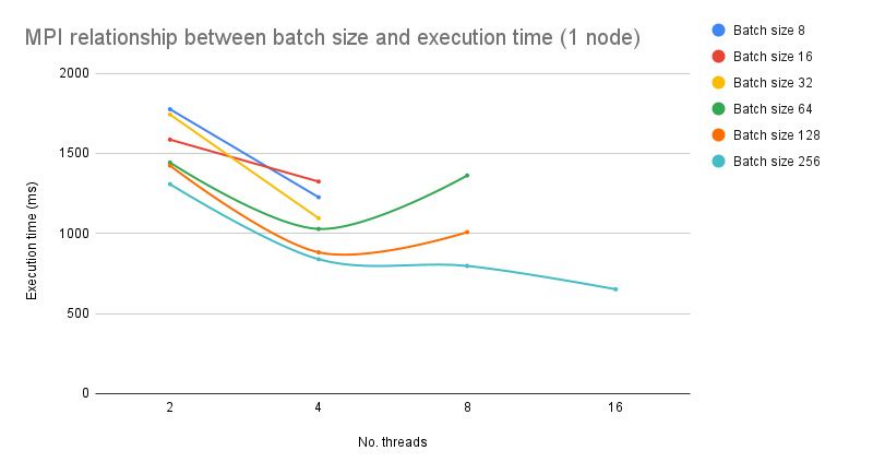


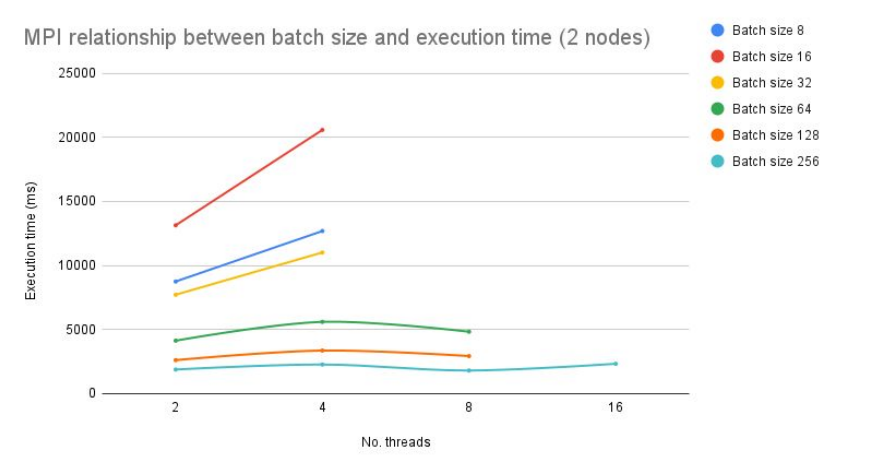
OpenMPI implementation

- Parallelization done by splitting into smaller problems
  - MPI_Scatterv to distribute the uneven load (matrix A)
  - MPI_Bcast to broadcast the matrix B
  - MPI_Gather to consolidate the results into matrix C
  - Other computations follow the same pattern (adding bias, hadamard product …)



OpenMPI benchmarking





- 1 node works faster than 2 nodes
- Increased batch size yields lower execution times

QA