1 Code profiling

Profiling my test code of one forward pass and one backprop pass, I get the following results for gpu kernel runtimes

- gemm gpu kernel - 86%
- row sum gpu kernel - 11%
- matrix transpose kernel - 0.8%
- matrix add kernel - 0.4%

As such it is clear that the most performance can be gained by optimizing the GEMM kernel. With my current implementation the GEMM operation is distributed across threads with each thread computing a single element of the final matrix, without using shared memory (the simplest suggested method). As such using a more sophisticated block GEMM kernel would likely improve the performance.

Once the GEMM has been optimized, further performance gains could be made by optimizing the row sum kernel. Additionally further improvements could be made by making more sophisticated use of MPI, such as distributing the network weights better.