Optimization of neural computations using functional data-parallel language

Naums Mogers
naums.mogers@ed.ac.uk
www.naumsmogers.me

Supervisors:
Christophe Dubach
Mike O'Boyle
Kenneth Heafield

The problem

Modern neural networks are large and require a lot of computational power, which sequential systems are unable to provide.

Parallel systems that are used for neural training are hard to optimize due to an infinitely large optimizational space.

Hardware configurations vary and evolve rapidly, which harms code portability: manual approach is expensive, and machine learning libraries are not performance portable.

The approach

Encode nets using functional data-parallel language Lift
www.lift-lang.org

Rewrite rules encode fine-grained structural optimizations...

Introduce neural network-specific rewrite rules...

Generate hundreds of code variants

CPU FPGA GPU

Choose the target hardware platform

...and parameters for autotuning

...that approximate calculations and optimize network configuration

AI

Choose the best code variant for a target platform using machine learning

Generic Lift stack

Functional language

- Abstracted from hardware
- Algorithm-centred
- Pure and safe
- High-level, easy to use

Generic rewrite rules

- Preserve semantics
- Improve performance

- Vectorization
- Memory coalescing, tiling
- Blocking
- Expression simplification
- Mapping to ND threads
- Split kernels
- Share 32-bit registers
- Use proprietary subroutines

Domain-specific Lift stack

Additional constructs

- Encode information about neural network
- Specify required accuracy

Neural rewrite rules

- Alter semantics
- Improve performance
- Alter accuracy

- Float ops approximation
- Vary precision among layers
- Gradient quantization
- Layer number autotuning
- Layer size autotuning
- Training batch size autotuning
- Learning rate autotuning

OpenCL

- Low-level hardware management
- Parametrized — yields to autotuning
- Cross-platform

THE UNIVERSITY of EDINBURGH
informatics

EPSRC Centre for Doctoral Training in
Pervasive Parallelism