EPSRC Centre for Doctoral Training in Pervasive Parallelism

Optimization of neural computations in a functional data-parallel language

Naums Mogers

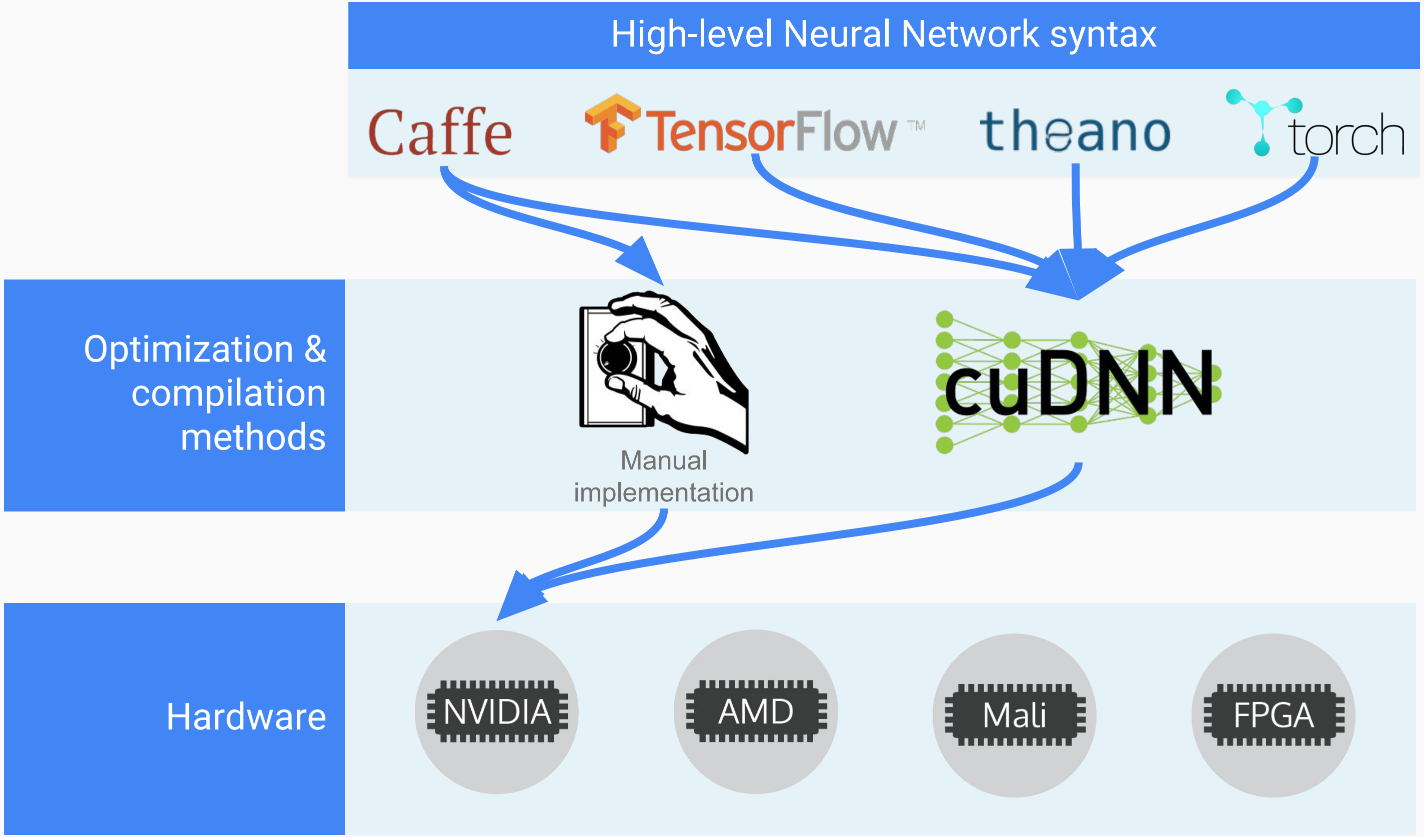
The problem

GPU code optimization: portability versus performance

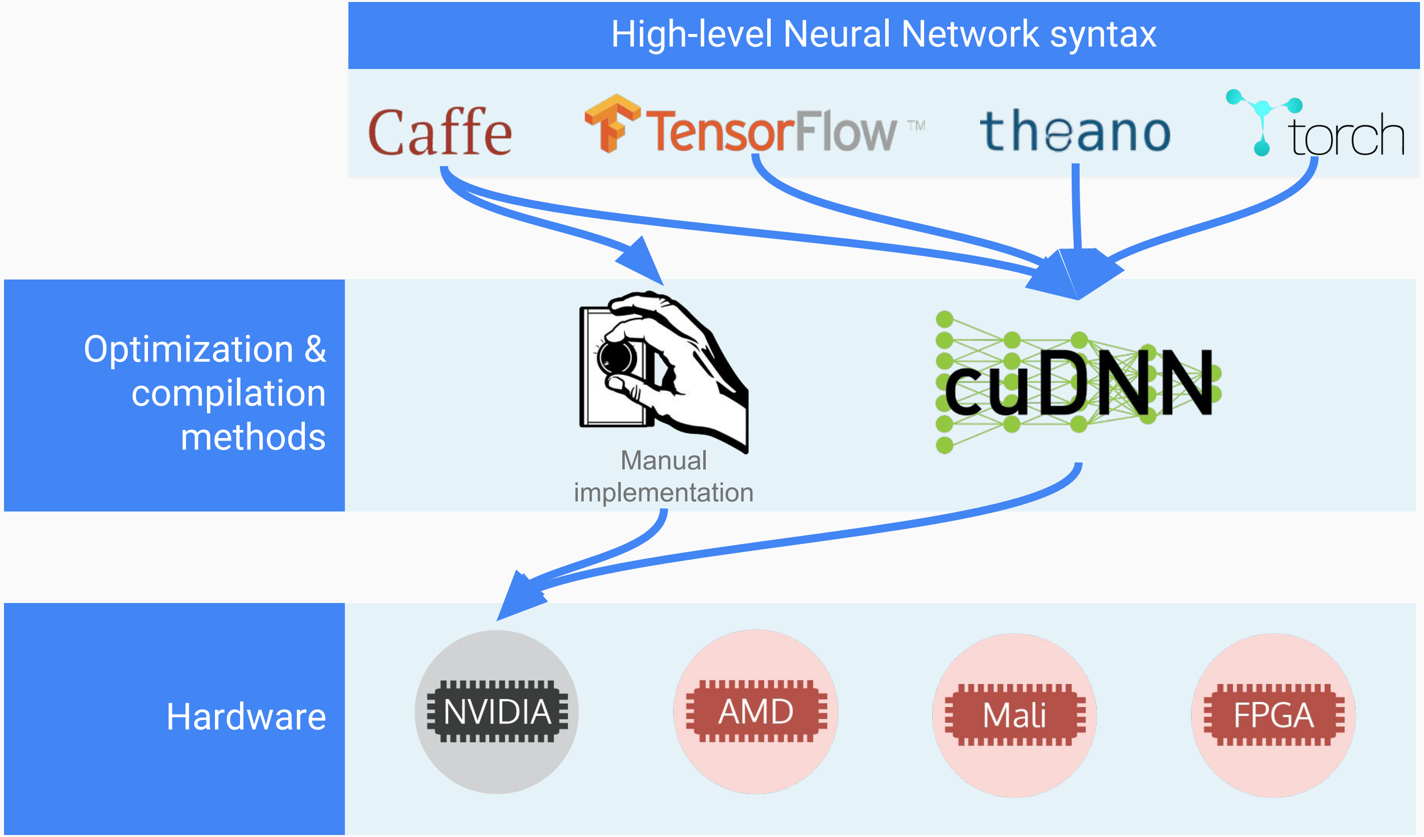
- Manual optimization → good performance
  → expensive to do
  → not portable
  → lack usability
  → does not support new devices

- Autotuners (PetaBricks, CLTune) → Functionally portable
  → not performance-portable
  → no structural optimizations

Existing approaches



Existing approaches



Existing approaches

The Lift language

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

Existing approaches

The Lift language

- Functional
  - Abstracted from hardware
  - Algorithm-centred
  - Pure and safe
  - High-level, easy to use
- Data-parallel
- Chooses the best OpenCL derivation for the target hardware
  - Both functionally and performance portable
  - Doesn't require hardware knowledge

Existing approaches

Lift’s rewrite rules

- Semantics-preserving transformations encoding fine-grained optimizations

Existing approaches

EPSRC Centre for Doctoral Training in
Pervasive Parallelism

Lift's rewrite rules

- Semantics-preserving transformations encoding fine-grained optimizations

Vectorization

Existing approaches

EPSRC Centre for Doctoral Training in
Pervasive Parallelism

Lift's rewrite rules

- Semantics-preserving transformations encoding fine-grained optimizations

Vectorization

Memory tiling

Existing approaches

Lift's rewrite rules

- Semantics-preserving transformations encoding fine-grained optimizations

Vectorization

Memory coalescing

Memory tiling

Existing approaches

Lift's rewrite rules

- Semantics-preserving transformations encoding fine-grained optimizations

Vectorization

Blocking

Memory coalescing

Memory tiling

Existing approaches

Lift's rewrite rules

- Semantics-preserving transformations encoding fine-grained optimizations

Vectorization

Blocking

ND mapping

Memory coalescing

Memory tiling

Existing approaches

EPSRC Centre for Doctoral Training in Pervasive Parallelism

Lift's rewrite rules

- Semantics-preserving transformations encoding fine-grained optimizations

Vectorization

Blocking

ND mapping

Memory coalescing

Memory tiling

Simplification

The method:

Neural Network-specific extension of the Lift language

The method

The extension:

- Neural Network (NN)-specific optimizations

The method

The user

- Encodes the NN in Lift
- Specifies the minimum required accuracy

The method

EPSRC Centre for Doctoral Training in Pervasive Parallelism

The user

- Encodes the NN in Lift
- Specifies the minimum required accuracy

Lift

- Applies generic optimizations
- Optimizes the NN code without preserving semantics
- Abides by the required accuracy

The method

Proposed optimizations:

- Approximations
  - Floating operations
  - Different layer precisions
  - Gradient quantization

The method

Proposed optimizations:

- Approximations
  - Floating operations
  - Different layer precisions
  - Gradient quantization

- NN configuration autotuning
  - Layer number
  - Layer size
  - Training batch size
  - Learning rate

Evaluation

The extension will be evaluated by

- Implementing Convolutional Neural Network (CNN) forward-propagation and training in Lift

Evaluation

EPSRC Centre for Doctoral Training in
Pervasive Parallelism

The extension will be evaluated by

- Implementing Convolutional Neural Network (CNN) forward-propagation and training in Lift
- Comparing CNN performance in domain-specific Lift vs generic Lift

Evaluation

The extension will be evaluated by

- Implementing Convolutional Neural Network (CNN) forward-propagation and training in Lift
- Comparing CNN performance in domain-specific Lift vs generic Lift
- Comparing CNN performance in domain-specific Lift vs Caffe

Evaluation

Evaluation metrics

- Forward-propagation runtime
- Training runtime
- The range of platforms supported

Conclusion

- Current GPU optimization methods are not performance portable
- Lift approach is performance portable
- Extend Lift to Neural Network-specific optimizations