MagmaDNN: Accelerated Deep Learning Using MAGMA

D. Nichols[1], K. Wong[1], S. Tomov[1], L. Ng[2], S. Chen[2], A. Gessinger[3]

1 - University of Tennessee, Knoxville
2 - The Chinese University of Hong Kong
3 - Slippery Rock University

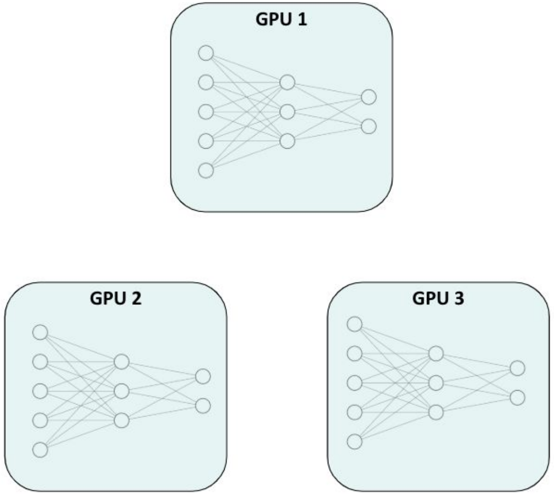
THE UNIVERSITY OF TENNESSEE KNOXVILLE

Organization

- Motivation
- Magma
- MagmaDNN Overview
  - Framework Overview
  - Compute Graph Optimization
  - Tuning
  - Distributed Training
- Results
- Current and Future Work
- Availability

Motivation

- Utilize state of the art MAGMA LA framework
- Provide a modular C++ Deep Learning Interface
- Support state of the art distributed training techniques



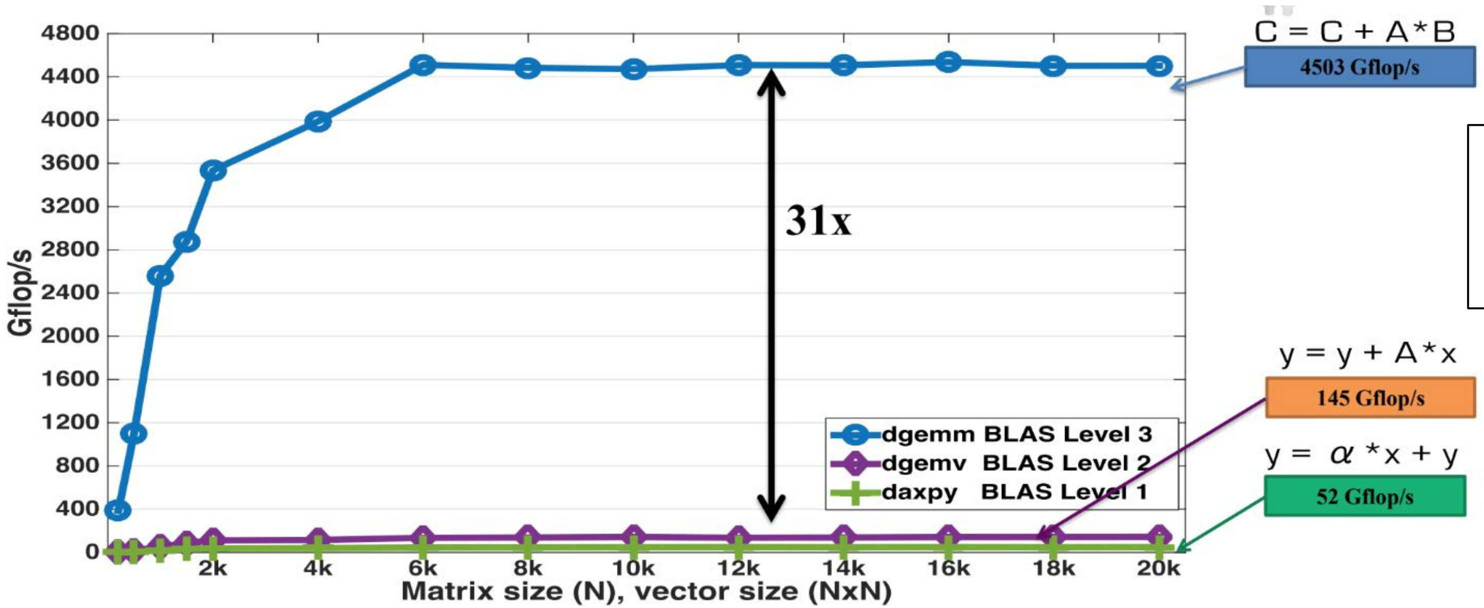
THE UNIVERSITY OF TENNESSEE KNOXVILLE

MAGMA

- Accelerated Linear Algebra on Heterogeneous Architectures

$$g_n\left(U_{n-1}W_n + b_n\right)$$

$$Y = A^T\left[\left[GgG^T\right]\odot\left[B^T dB\right]\right]A$$



P100 1.19GHz

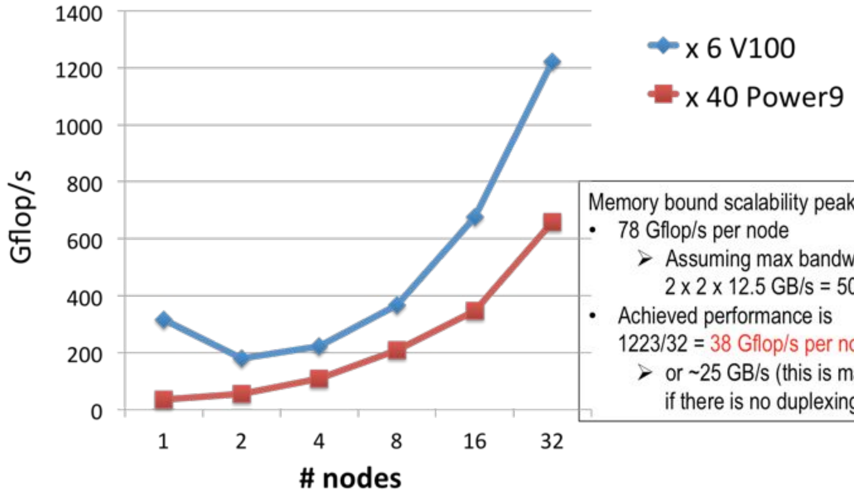
Peak: 4700 Gflops

THE UNIVERSITY OF TENNESSEE KNOXVILLE

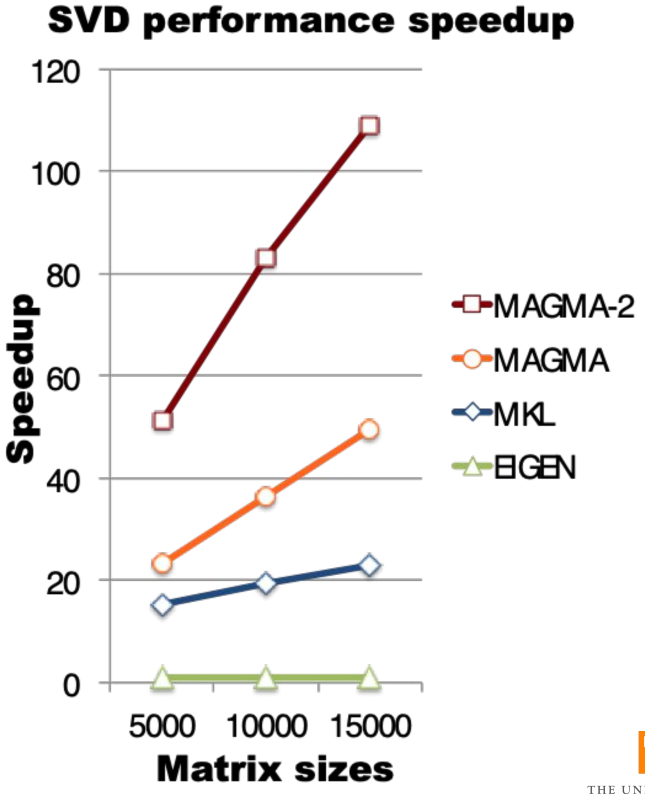
MAGMA for Data Science

- Better performance than other leading LA packages in SVD
- Scalable FFTs for Convolutions

Strong scalability of 3D FFT on Summit (N = 1024)

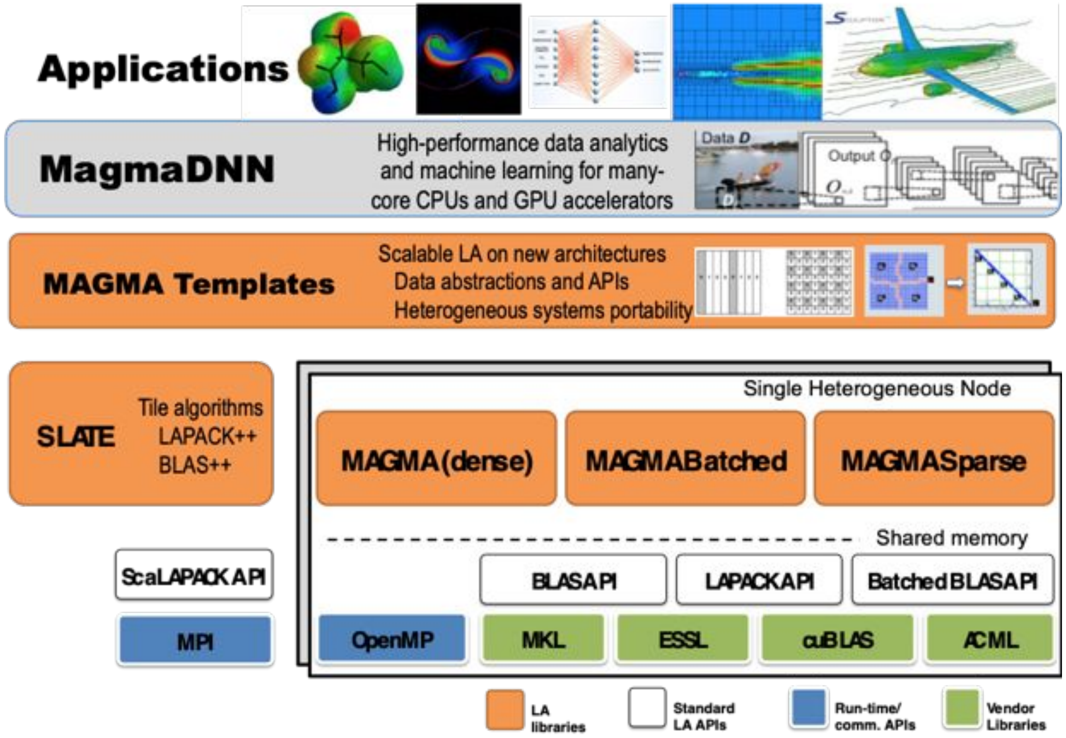


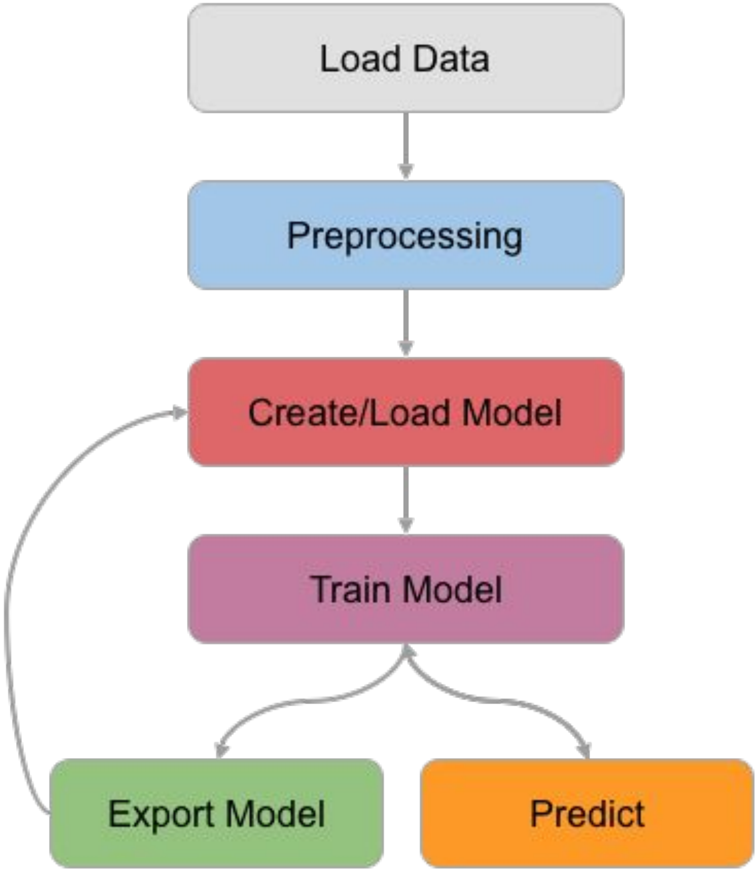
Memory bound scalability peak:
- 78 Gflop/s per node
  - Assuming max bandwidth 2 x 2 x 12.5 GB/s = 50 GB/s
- Achieved performance is 1223/32 = 38 Gflop/s per node
  - or ~25 GB/s (this is maximum if there is no duplexing)



THE UNIVERSITY OF TENNESSEE KNOXVILLE

MagmaDNN Overview





Framework Overview

Memory Manager

Tensor

Operation & Graph

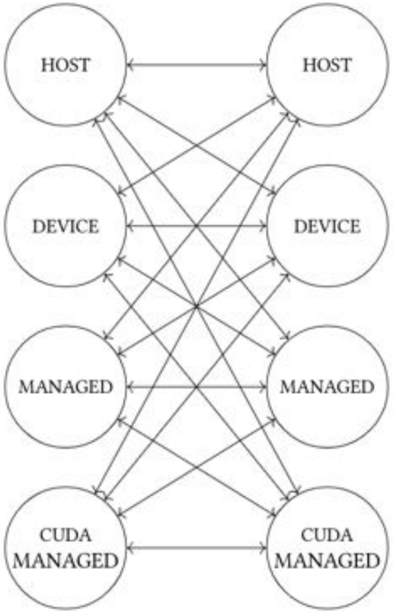
Optimizer

Model

Framework Overview

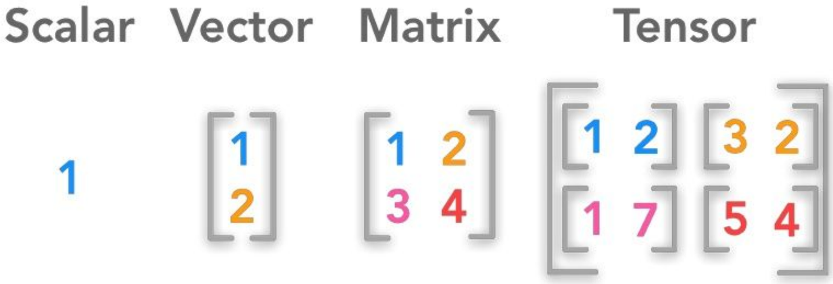
Memory Manager | Tensor | Operation & Graph | Optimizer | Model

```
size_t n_vals = 20;
// HOST, DEVICE, MANAGED, CUDA_MANAGED
memory_t mem_type = MANAGED;
device_t device_id = 0;
MemoryManager<float> mm(n_vals, mem_type, device_id);
```



Framework Overview

Memory Manager | **Tensor** | Operation & Graph | Optimizer | Model

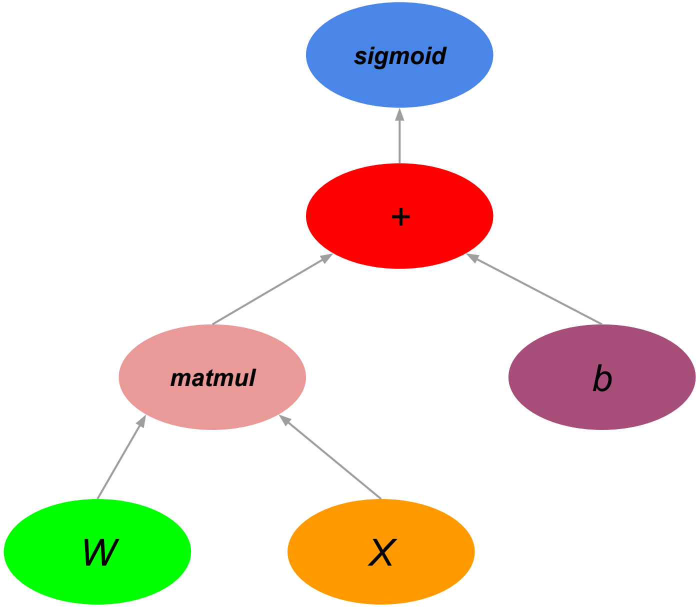



```
Tensor<float> x({32,28,28}, {UNIFORM, {-1.0f,1.0f}}, DEVICE);
std::cout << x.get({1,3,0});
x.set({1,3,0}, 8.0f);
```

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Framework Overview

Memory Manager | Tensor | Operation & Graph | Optimizer | Model




```
auto X = op::var<float>("X", {5,3},{UNIFORM});
auto W = op::var<float>("W", {6,5},{UNIFORM});
auto b = op::var<float>("b", {6,3},{UNIFORM});

auto transform = op::add( op::matmul(W, X), b );

Tensor<float> *output = transform->eval();
```

Framework Overview

Memory Manager | Tensor | Operation & Graph | **Optimizer** | Model

```cpp
auto x = op::var<float>("x", NONE);
auto c = op::var<float>("c", {CONSTANT, -2.0f});


optimizer::GradientDescent opt(0.05);


opt.minimize( op::add(op::pow(x, 2), c), {x});
```

minimize $x^2 + c$

with respect to *x*

Framework Overview

Memory Manager | Tensor | Operation & Graph | Optimizer | Model

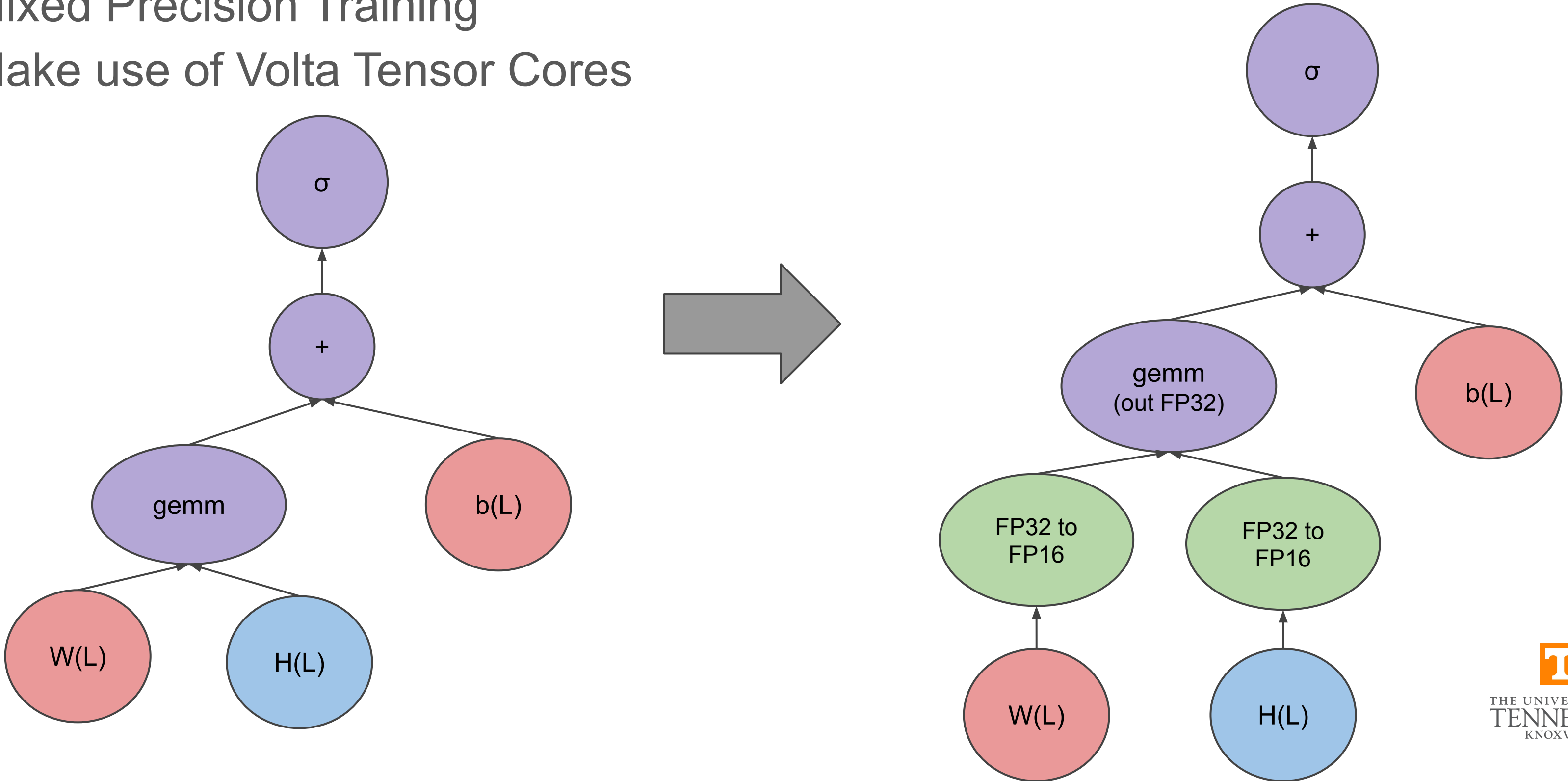
```
Tensor<float> data({60000, 28*28}, HOST);
io::read_csv_to_tensor(data, "mnist_data_set.csv");
Tensor<float> labels({60000, 10}, HOST);
io::read_csv_to_tensor(labels, "mnist_labels_set.csv");

/* create a vector of layers ... */

model::NeuralNetwork<float> model(layers_vector, optimizer::CROSS_ENTROPY, optimizer::ADAM,
{batch_size, n_epochs, learning_rate});
model.fit(data, labels, params_out, verbose);
```

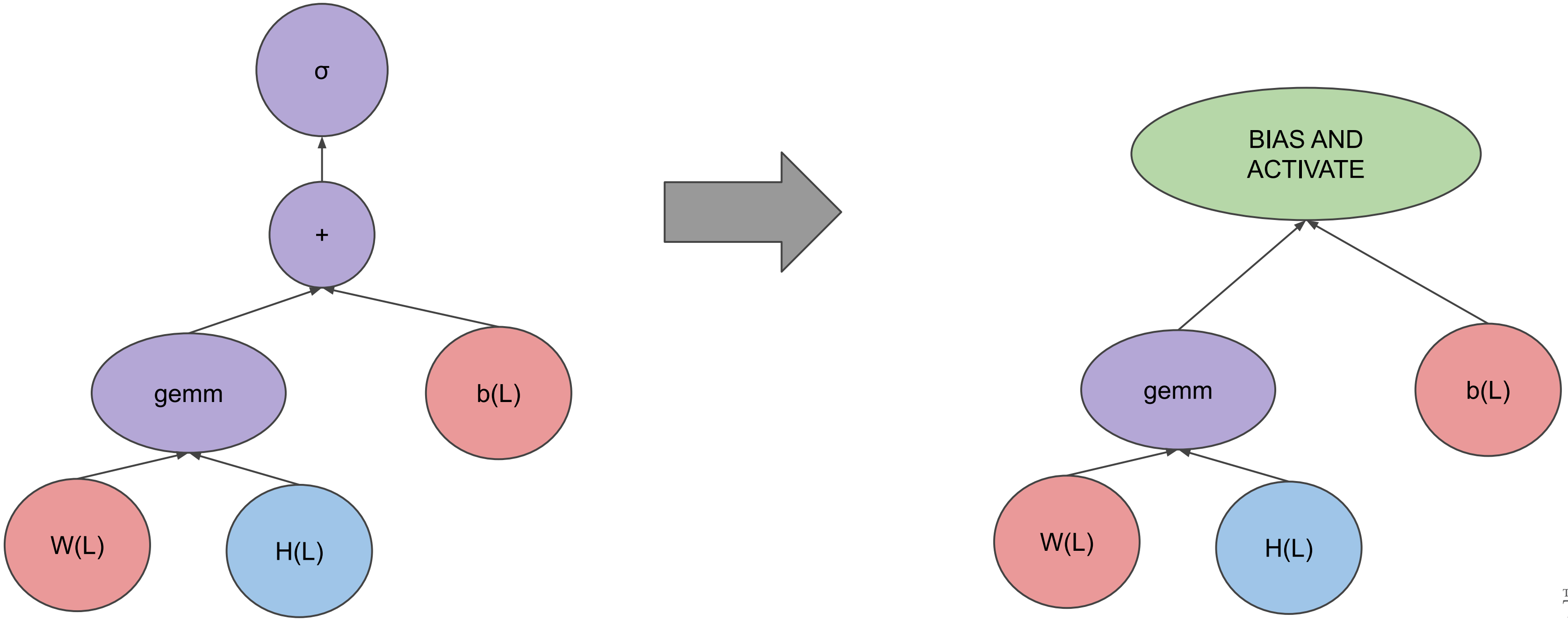
Compute Graph Optimization

- Mixed Precision Training
- Make use of Volta Tensor Cores



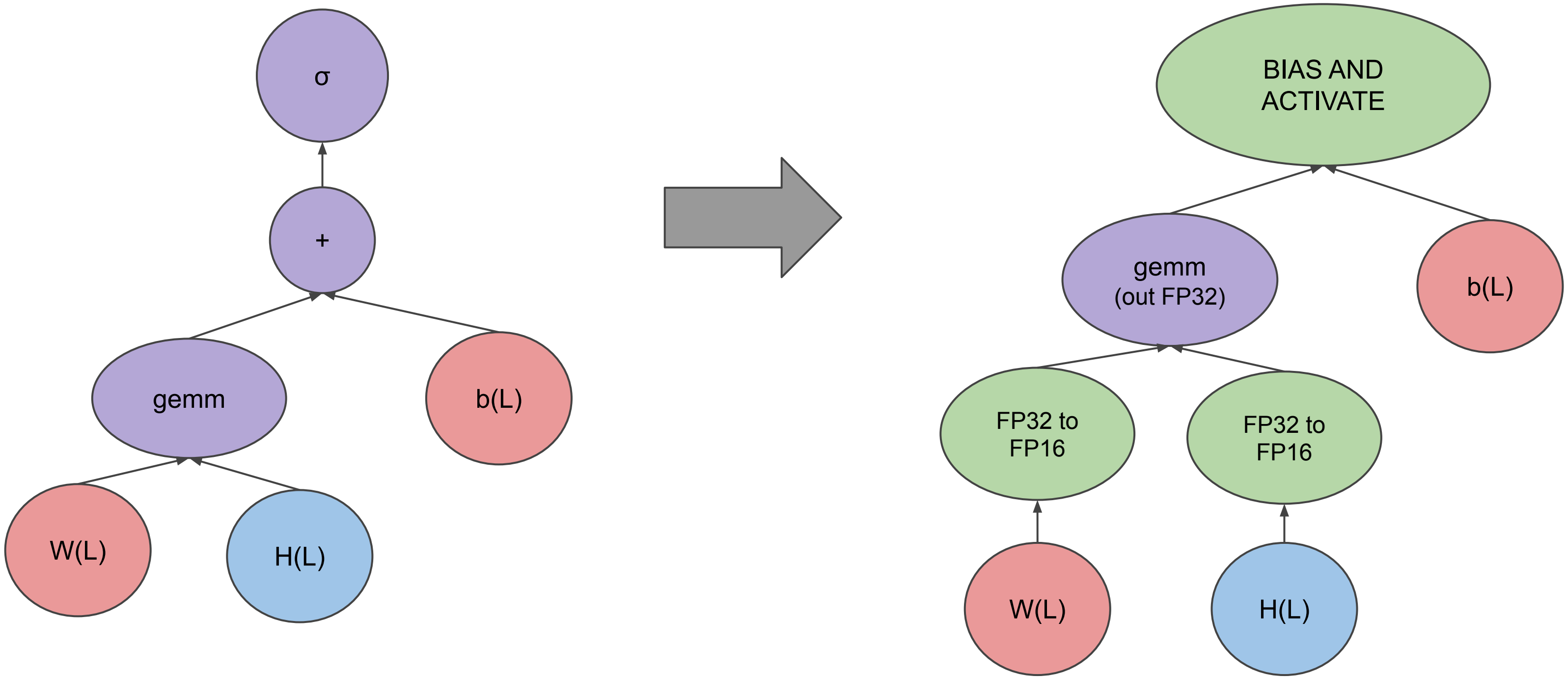
Compute Graph Optimization (cont.)

- Fused Operations



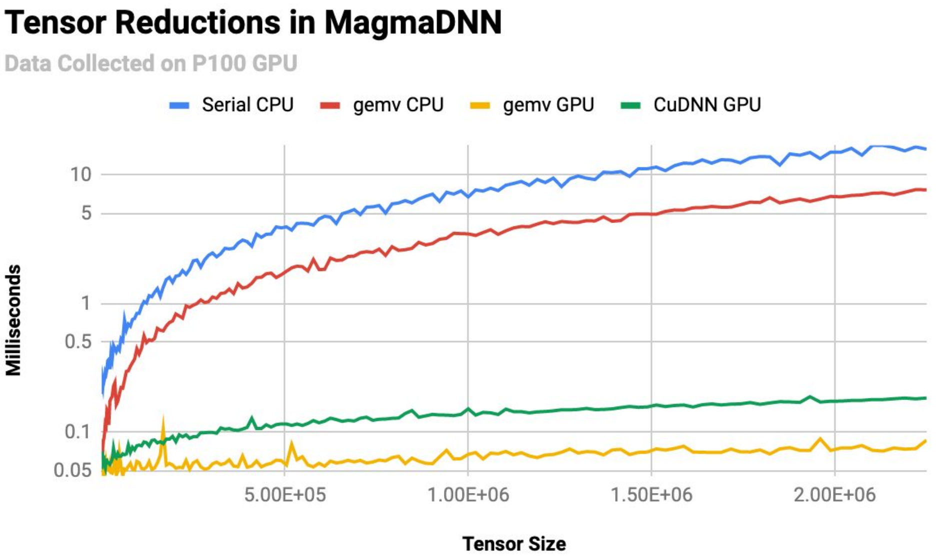
Compute Graph Optimization

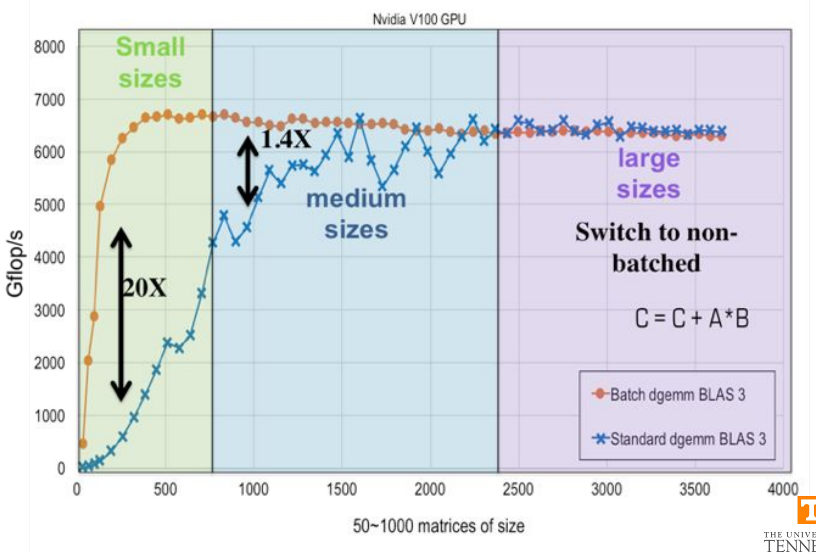
- Combined



Tuning

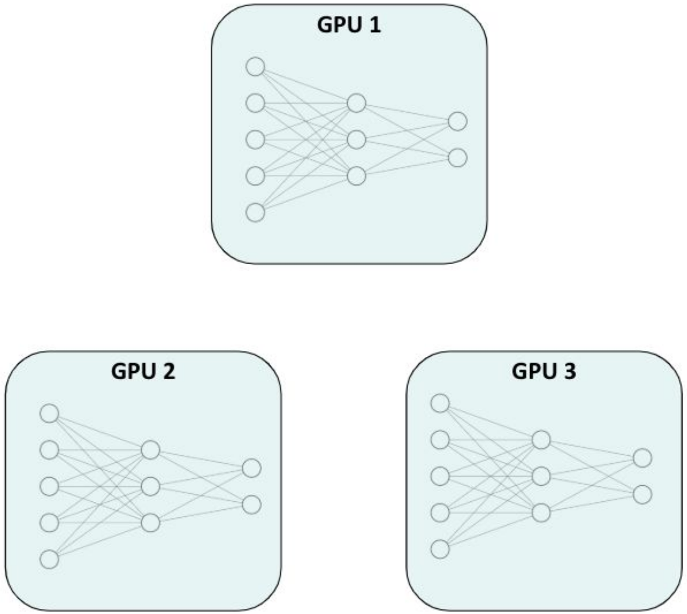
- MagmaDNN tunes tensor and deep learning kernels
- Magma tunes matrix algebra kernels



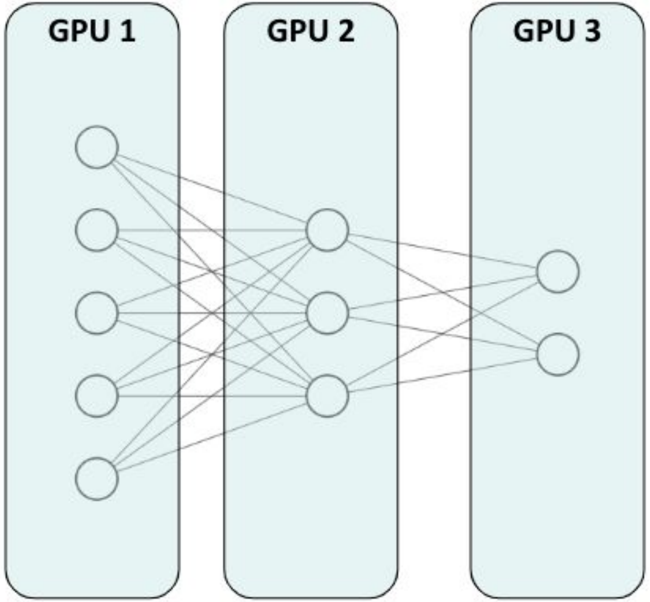


Distributed Training

Data Parallelism

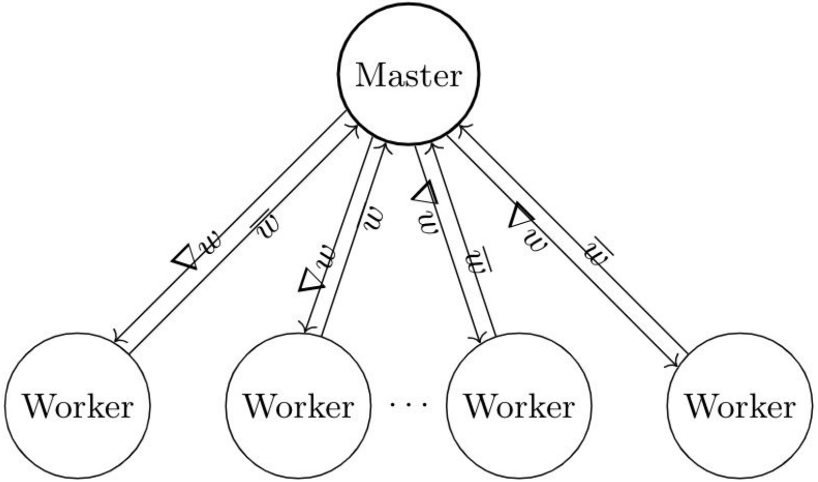


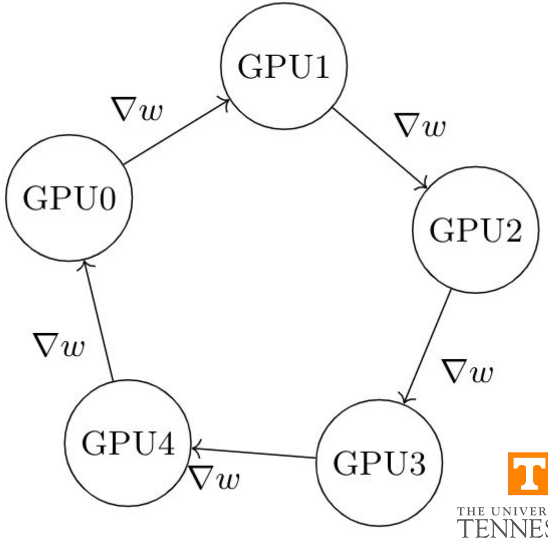
Model Parallelism



Distributed Training

- ASGD
- AllReduce
- Ring Reduce

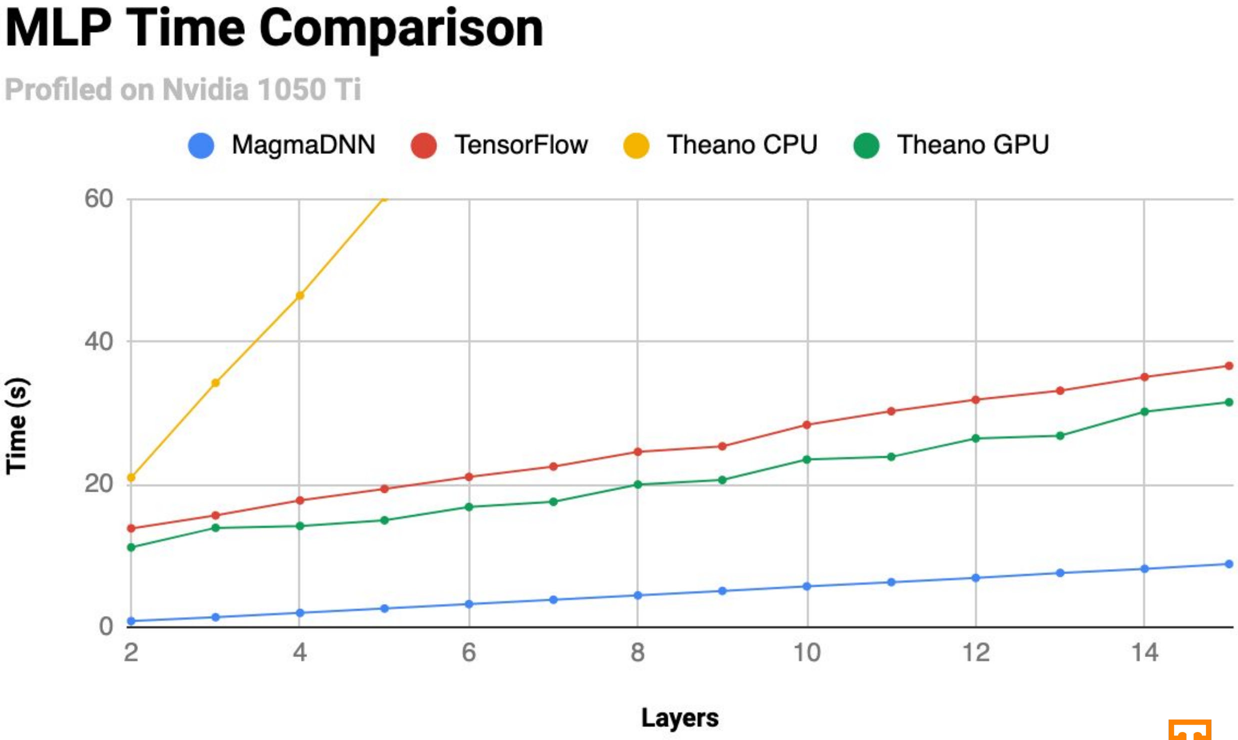




THE UNIVERSITY OF TENNESSEE KNOXVILLE

Results

- Best performance across a single node
- Best scaling with network size



Summary

- Accelerated single node training times
- Modern C++ interface
- Support for distributed training

Current and Future Work

- Competitive distributed ResNet-50 training time
- Move to new C++ standard
- More “Bells and Whistles”
- Development
- Python Interface

Acknowledgements

- Joint Institute for Computer Science (JICS)
- National Institute for Computational Sciences (NICS)
- University of Tennessee, Knoxville (UTK)
- NSF Award #1659502
- Extreme Science and Engineering Discovery Environment (XSEDE)

- BP High Performance Computing

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Availability

Development: https://github.com/MagmaDNN/magmadnn

Releases: https://bitbucket.org/icl/magmadnn

Tutorials: https://github.com/MagmaDNN/magmadnn/tree/master/docs/tutorials

Contact: Daniel Nichols (danielnichols@utk.edu)