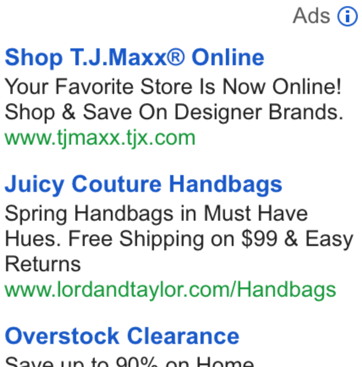
Deep Learning on GPU Clusters

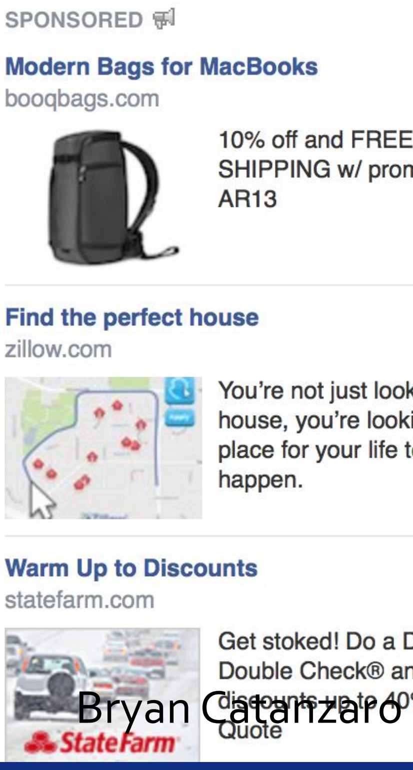
Bryan Catanzaro

Baidu Research

Machine Learning

- ML runs many things these days
  - Ad placement / Product Recommendations
  - Web search / image search
  - Speech recognition / machine translation
  - Autonomous driving
  - …



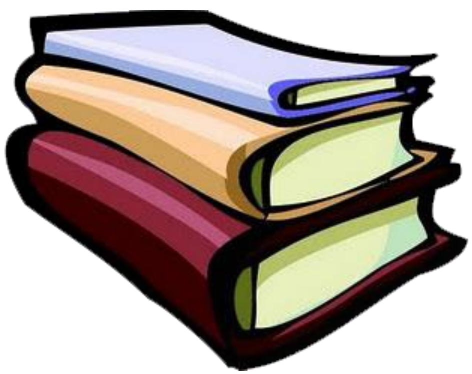


Machine learning in practice

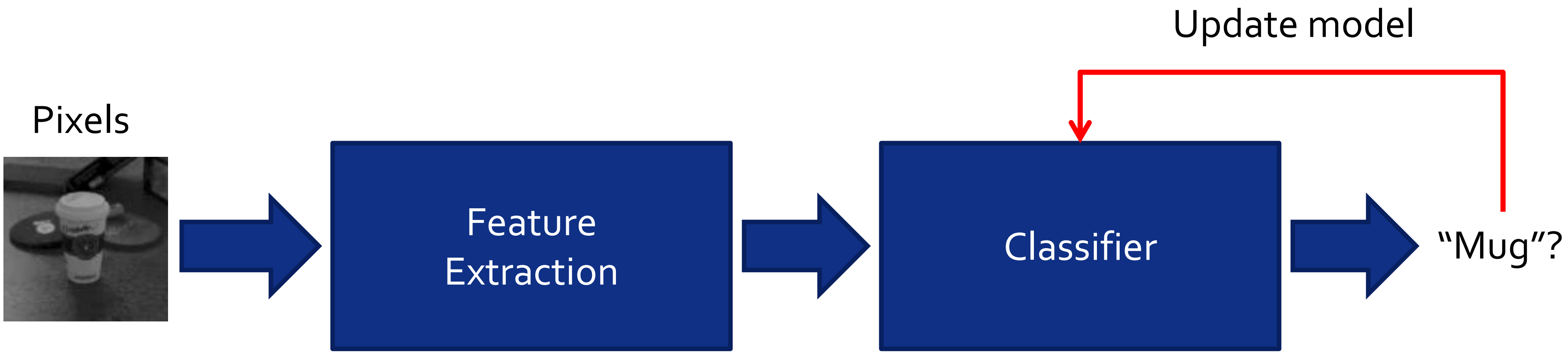
"Mug"

Cylinders; handles

Machine Learning (Classifier)

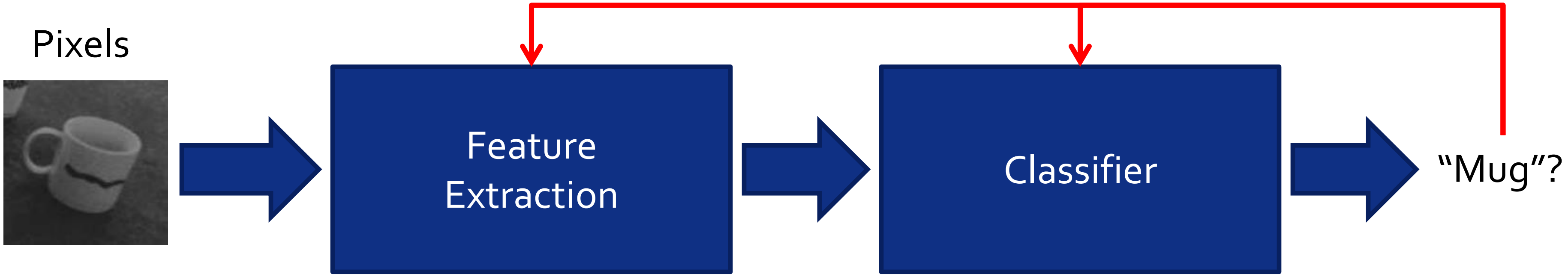


How does ML work?



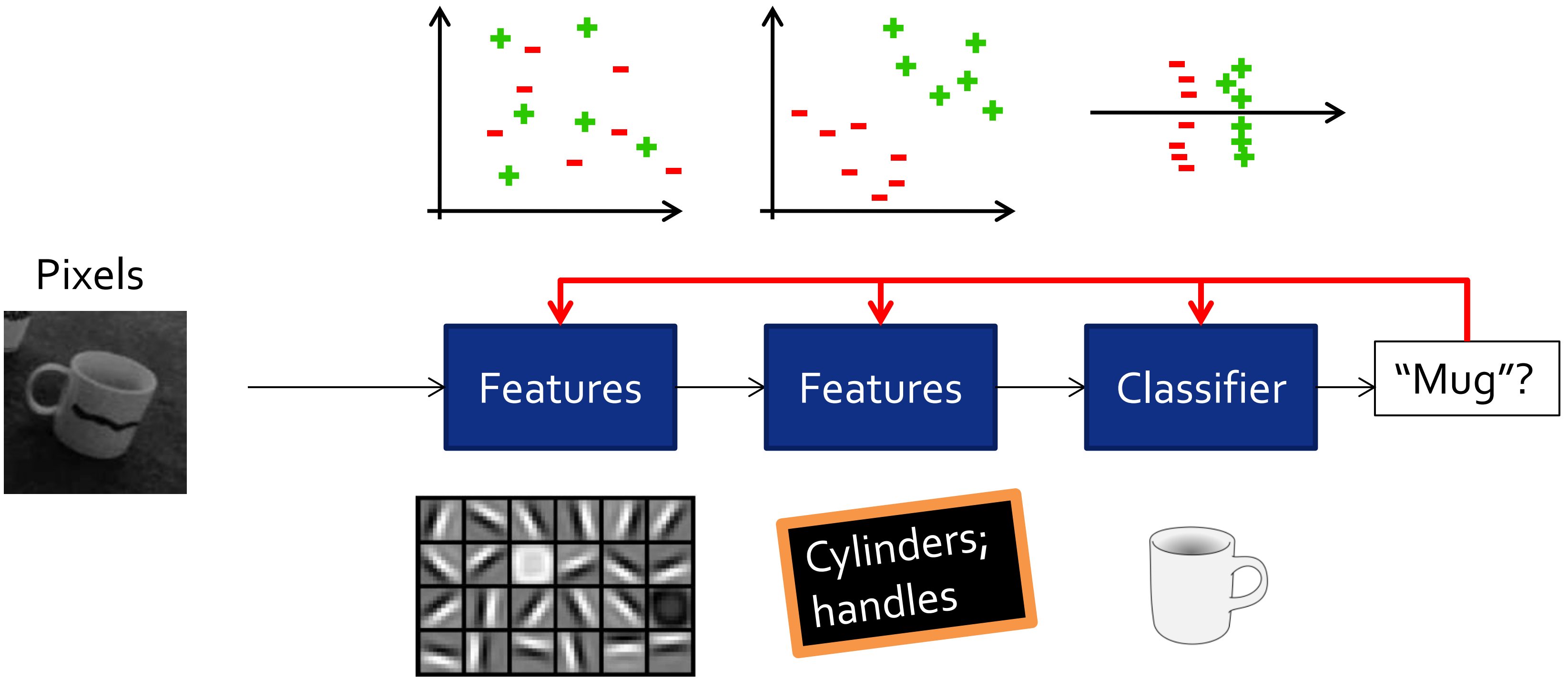
Learning features

- Can we learn “features” from data?



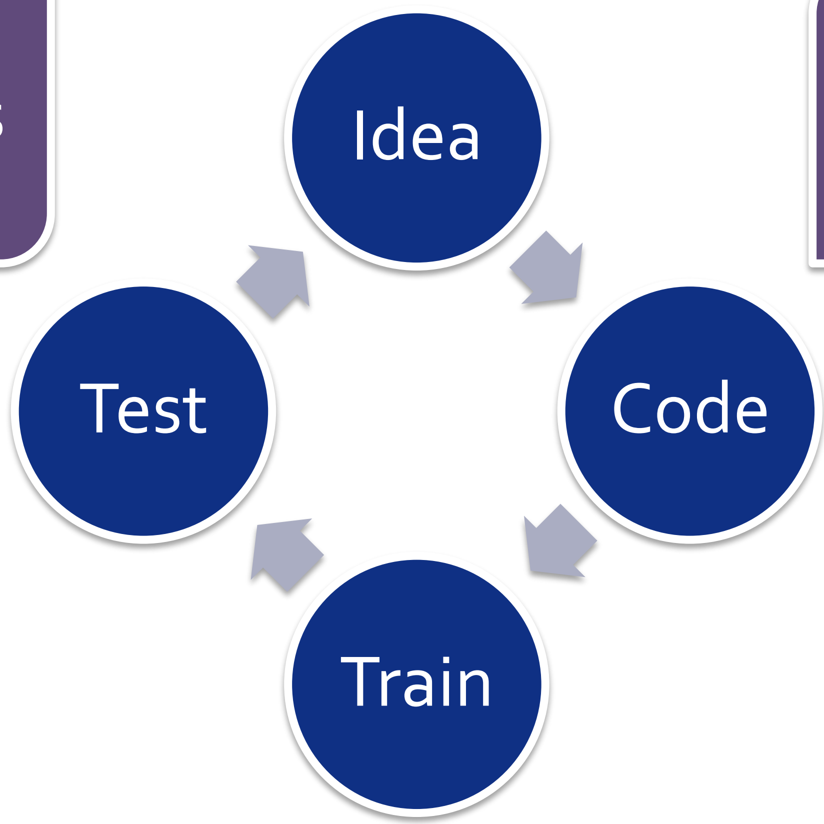
Learning features

- Deep learning: learn multiple stages of features to achieve end goal.



Progress in AI

Iteration latency gates progress



Fastest time from idea to tested model

Deep Learning brings special opportunities and challenges

Adam Coates

The need for scaling

- Dist-Belief [Le et al., ICML 2012]: Up to 1.7 billion parameter networks.
- Unsupervised learning algorithm with > 1 billion parameters able to discover “objects” in high-res images.

Faces: 

Cats: 

Bodies: 

➢ **1000 machines for 1 week. (16000 cores.)**

[Also: Dean et al., NIPS 2012]

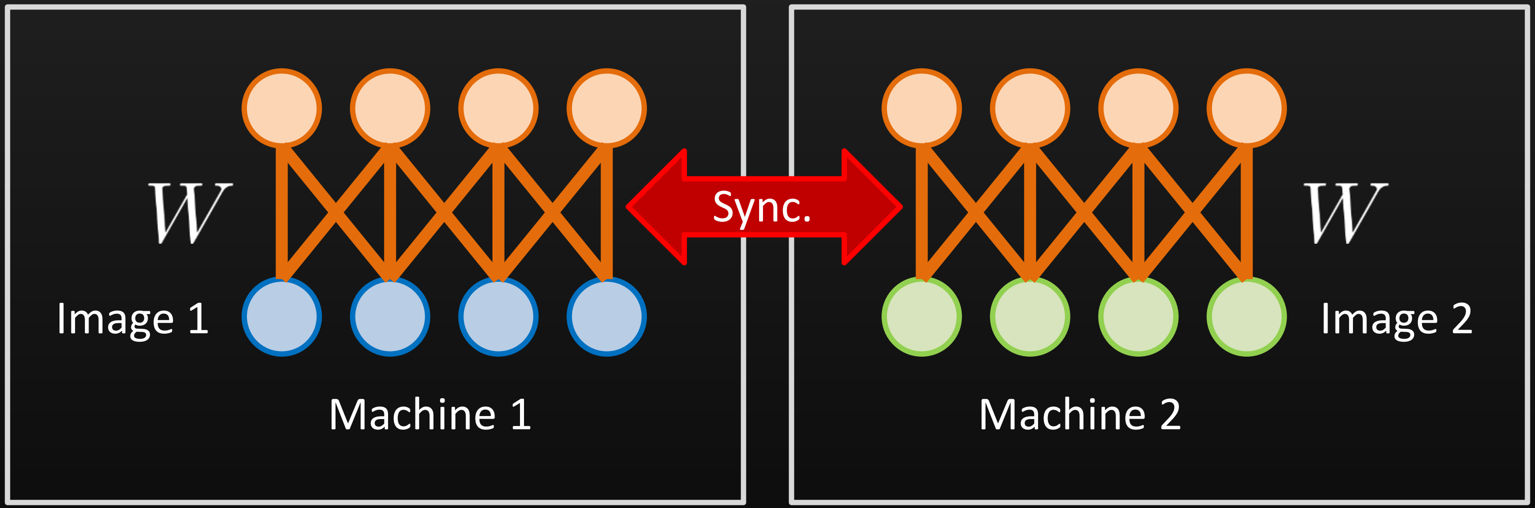
What will the rest of us do??

- Millions of $$ hardware.
- Extensive engineering to handle node failures and network traffic.
- Hard to scale beyond data center.
  - …if you had a data center.

Adam Coates, Brody Huval, Tao Wang, David J. Wu, Andrew Y. Ng and Bryan Catanzaro

Two ways to scale neural networks

- Simple solution: "data parallelism"
  - Parallelize over images in a batch.



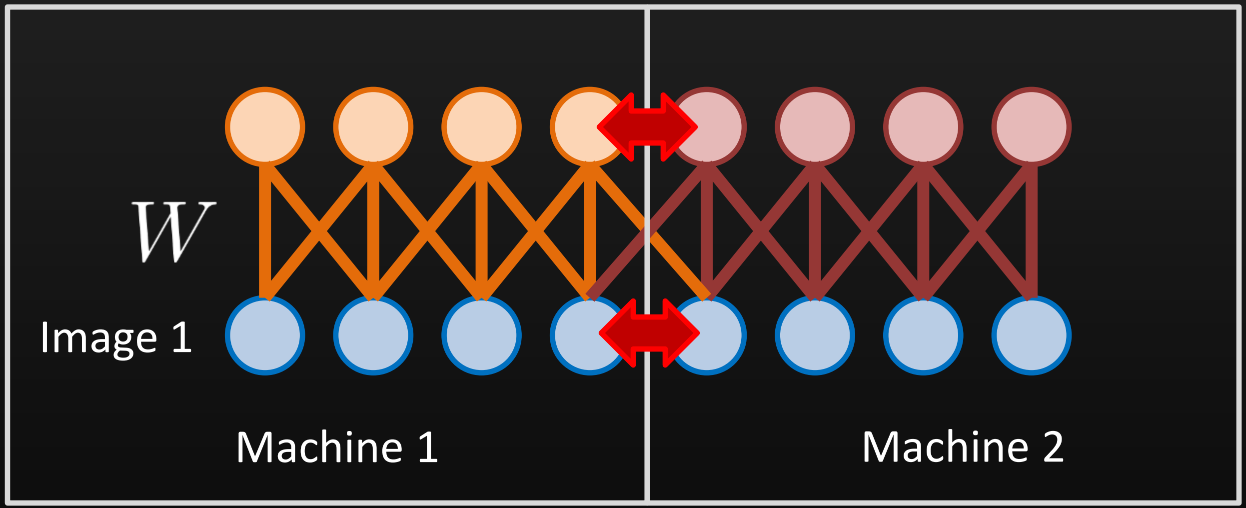
➢ Need to synchronize model across machines.

➢ Difficult to fit big models on GPUs.

Adam Coates, Brody Huval, Tao Wang, David J. Wu, Andrew Y. Ng and Bryan Catanzaro

Two ways to scale neural networks

- "Model parallelism"
  - Parallelize over neurons. (Relies on local connectivity.)



- ➢ Scales to much larger models.
- ➢ Much more frequent synchronization.

The network bottleneck

- On a typical Ethernet cluster:
  - Data parallelism:
    - Synchronize a 1B parameter model = **30 seconds**.
  - Model parallelism:
    - Move 1MB of neurons for 100 images = **0.8 seconds**
      - Must do this for *every* layer.
      - Typically >>10 times slower than computation.
- Problem: communication makes distribution very inefficient for large neural nets.
  - How do we scale out efficiently??

COTS HPC Hardware

- Infiniband:
  - – FDR Infiniband switch.
  - – 1 network adapter per server.

    56 Gbps; microsecond latency.

- GTX 680 GPUs
  - – 4 GPUs per server.

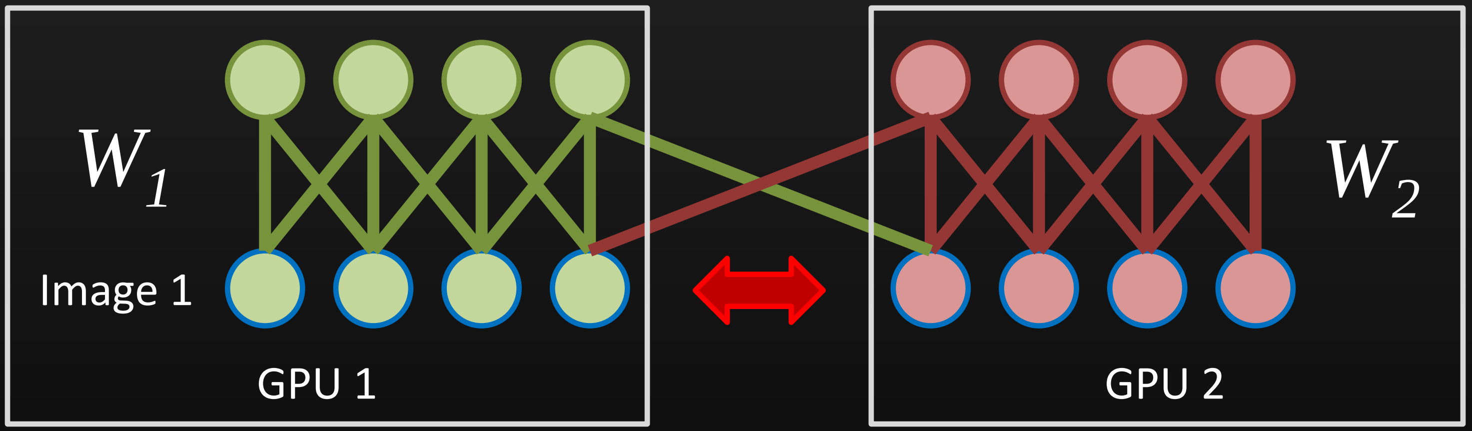
    > 1 TFLOPS each for ideal workload.

Adam Coates, Brody Huval, Tao Wang, David J. Wu, Andrew Y. Ng and Bryan Catanzaro

Model parallelism in MPI

MPI starts a single process for each GPU.

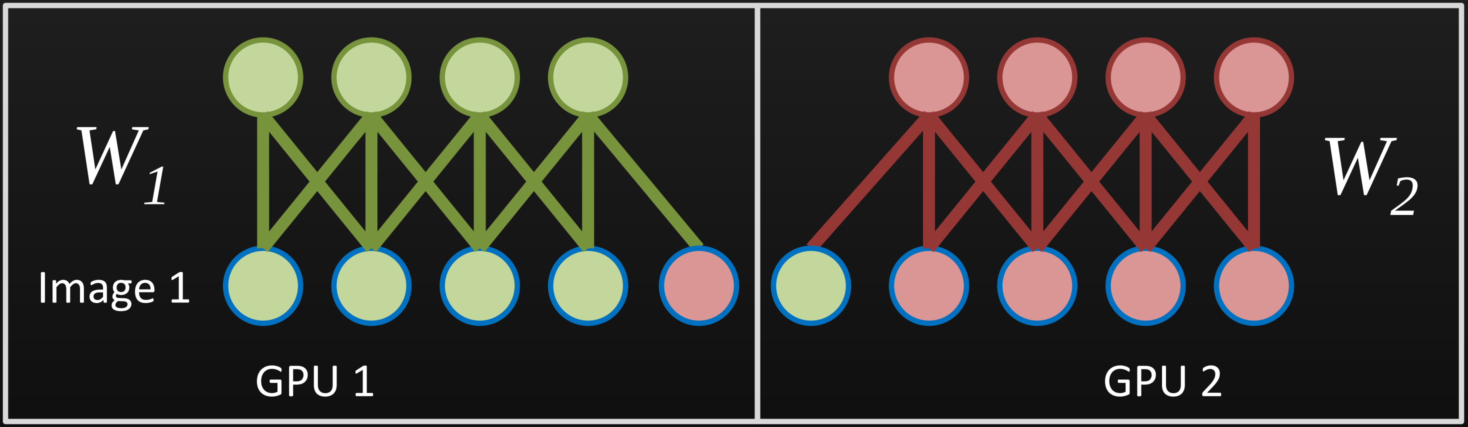
- Enables message passing, but this is surprisingly unnatural.



Model parallelism in MPI

MPI starts a single process for each GPU.

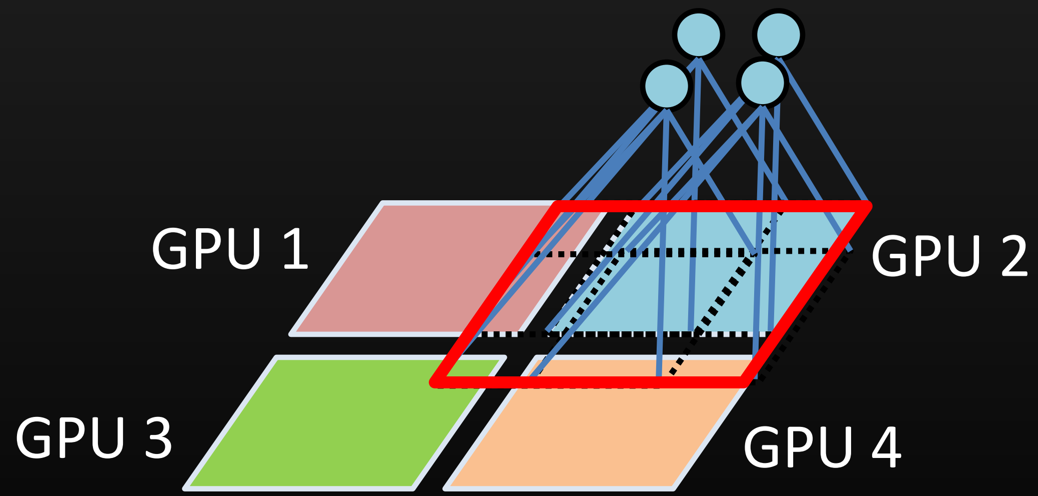
– Enables message passing, but this is surprisingly unnatural.



Adam Coates, Brody Huval, Tao Wang, David J. Wu, Andrew Y. Ng and Bryan Catanzaro

HPC Software Infrastructure: Communication

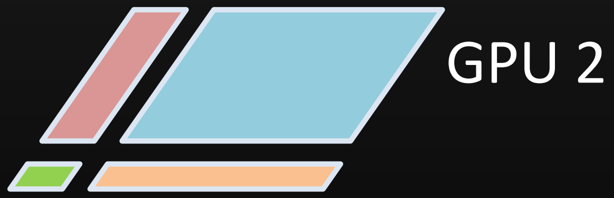
- Moving neuron responses around is confusing.
  - Hide communication inside “distributed array”.



Adam Coates, Brody Huval, Tao Wang, David J. Wu, Andrew Y. Ng and Bryan Catanzaro

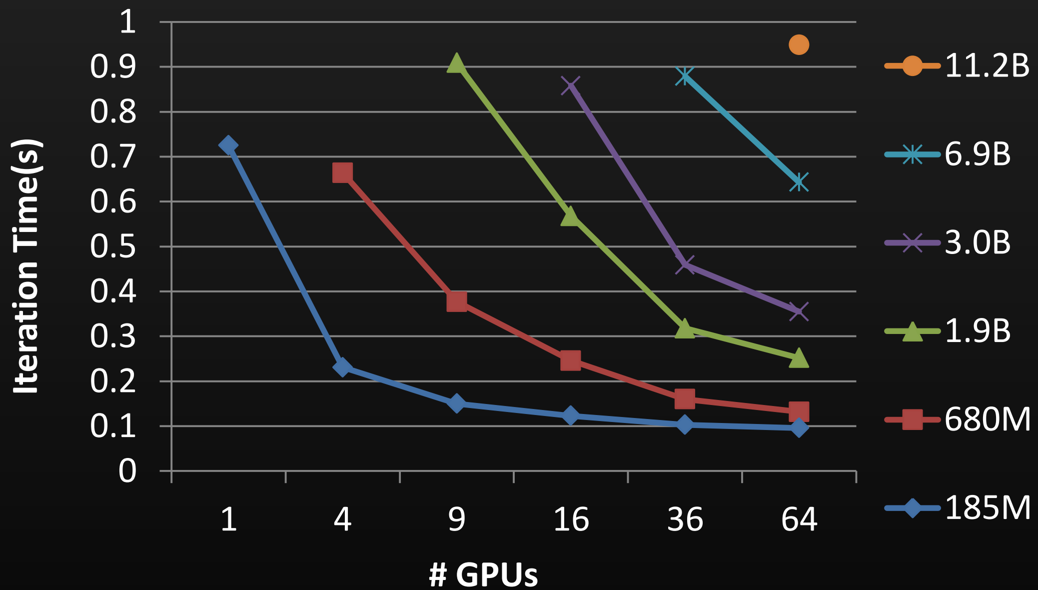
HPC Software Infrastructure: Communication

- After some hidden communication, GPU 2 has all the input data it needs.
  - GPU code not much different from 1 GPU.



Results: Scaling

- Implemented 9-layer network from Le et al., 2012.
  - 3 stacks of sparse auto-encoder, pooling, LCN.
  - Compute “fine-tuning” gradient.

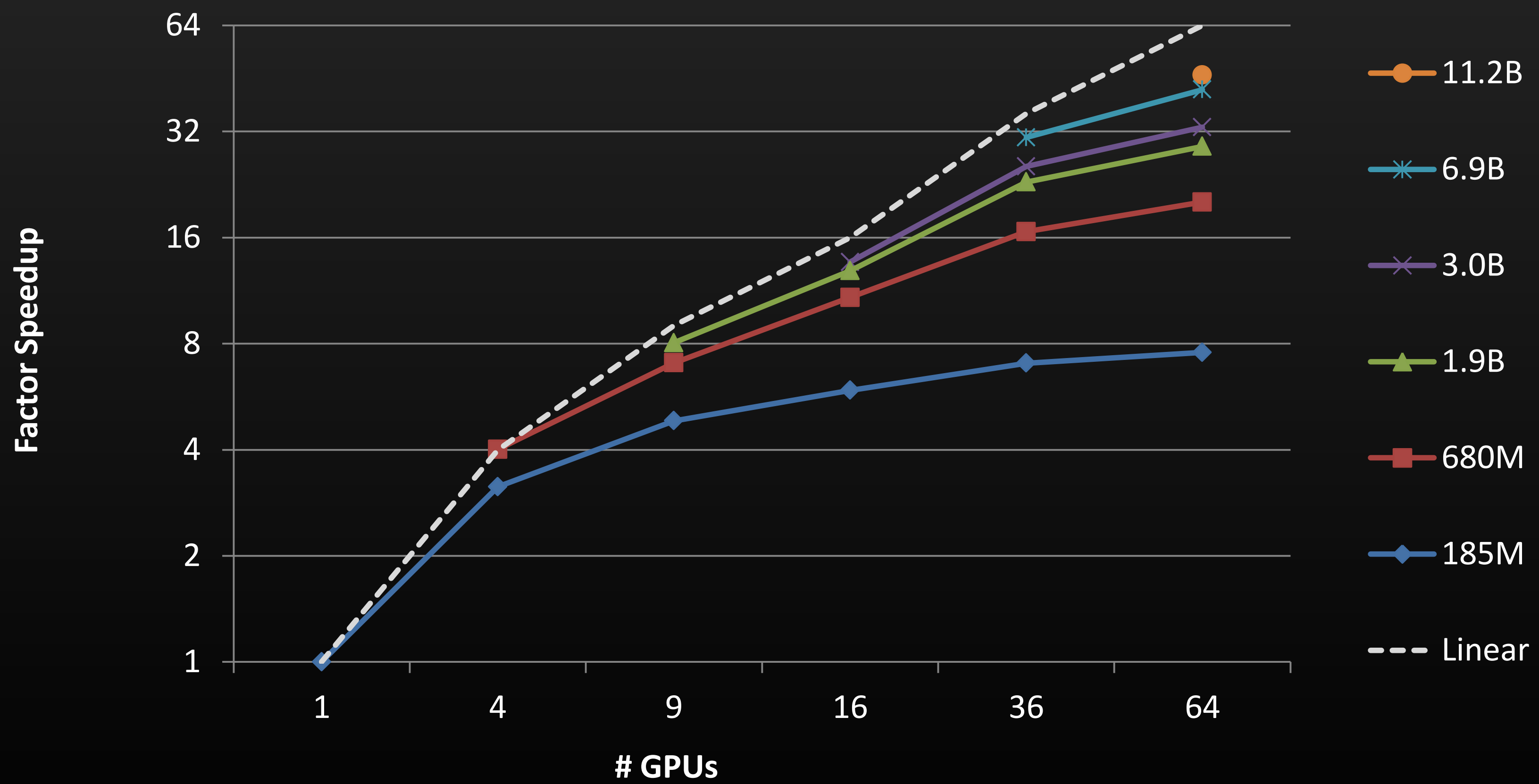


- Up to **11.2B** parameter networks.
  - Update time similar to **185M** parameter network on 1 GPU.

Adam Coates, Brody Huval, Tao Wang, David J. Wu, Andrew Y. Ng and Bryan Catanzaro

Results: Scaling

- Up to 47x increase in throughput:



Adam Coates, Brody Huval, Tao Wang, David J. Wu, Andrew Y. Ng and Bryan Catanzaro

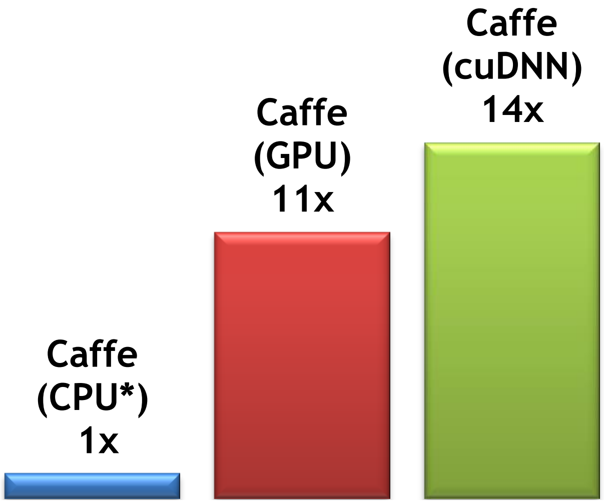
cuDNN

- Deep Neural Networks rely heavily on BLAS
  - Basic Linear Algebra Subroutines
- However, there are some kernels unique to DNNs
  - Such as convolutions
- cuDNN is a GPU library that provides these kernels
- Available at https://developer.nvidia.com/cudnn

Using Caffe with cuDNN

- Accelerate Caffe layer types by 1.2 – 3X
- On average, 36% faster overall for training on Alexnet
- Integrated into Caffe dev branch
  - Official release soon

Overall AlexNet training time



Baseline Caffe compared to Caffe accelerated by cuDNN on K40

***CPU is 24 core E5-2697v2 @ 2.4GHz Intel MKL 11.1.3**

Conclusion

- Deep Learning is increasingly important to AI
- HPC is key to Deep Learning

- Interested in applying your HPC skills to AI? Talk to us!

bcatanzaro@baidu.com