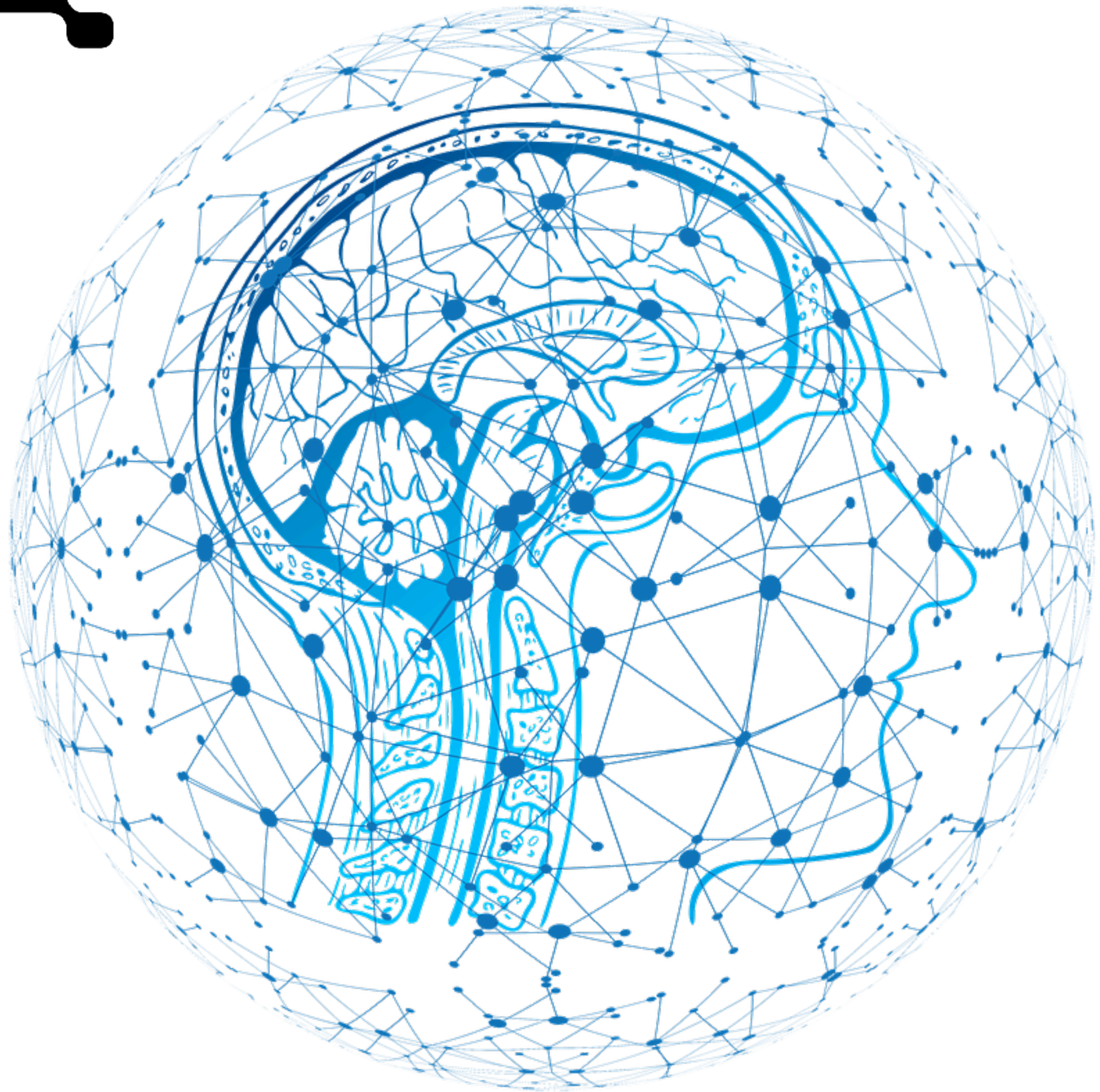
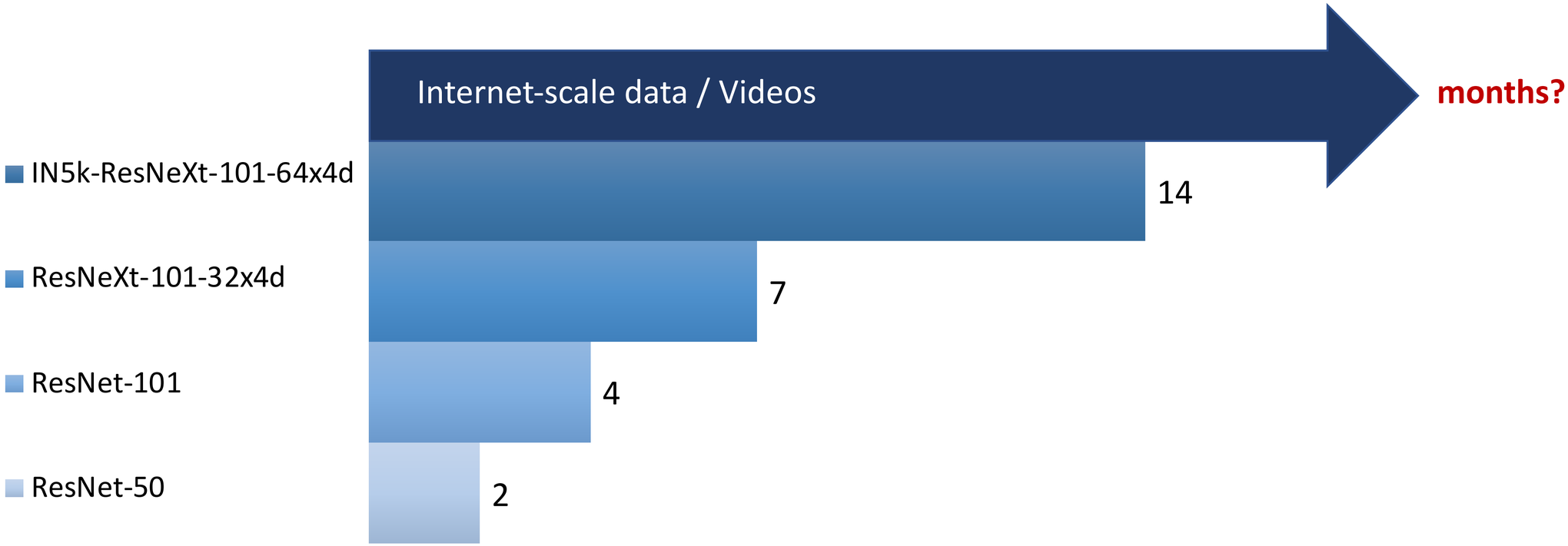


Introduction to High-performance Machine learning

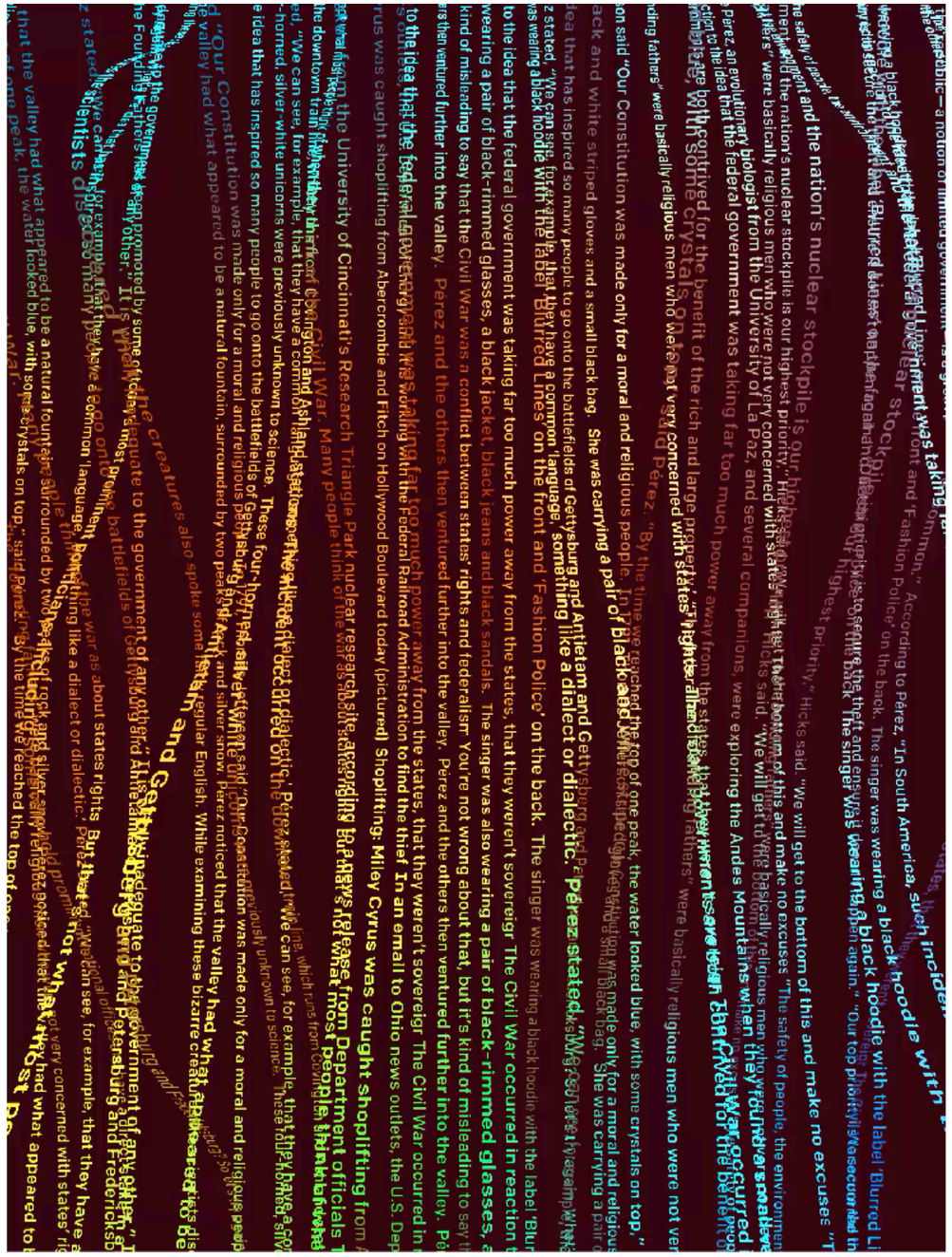
Maxwell Cai (SURFsara)

Training time (days)*



*measured on NVIDIA M40 8-gpus

facebook research



Credit: OpenAI

OpenAI’s GPT-2 has 1.5 billion parameters

Input

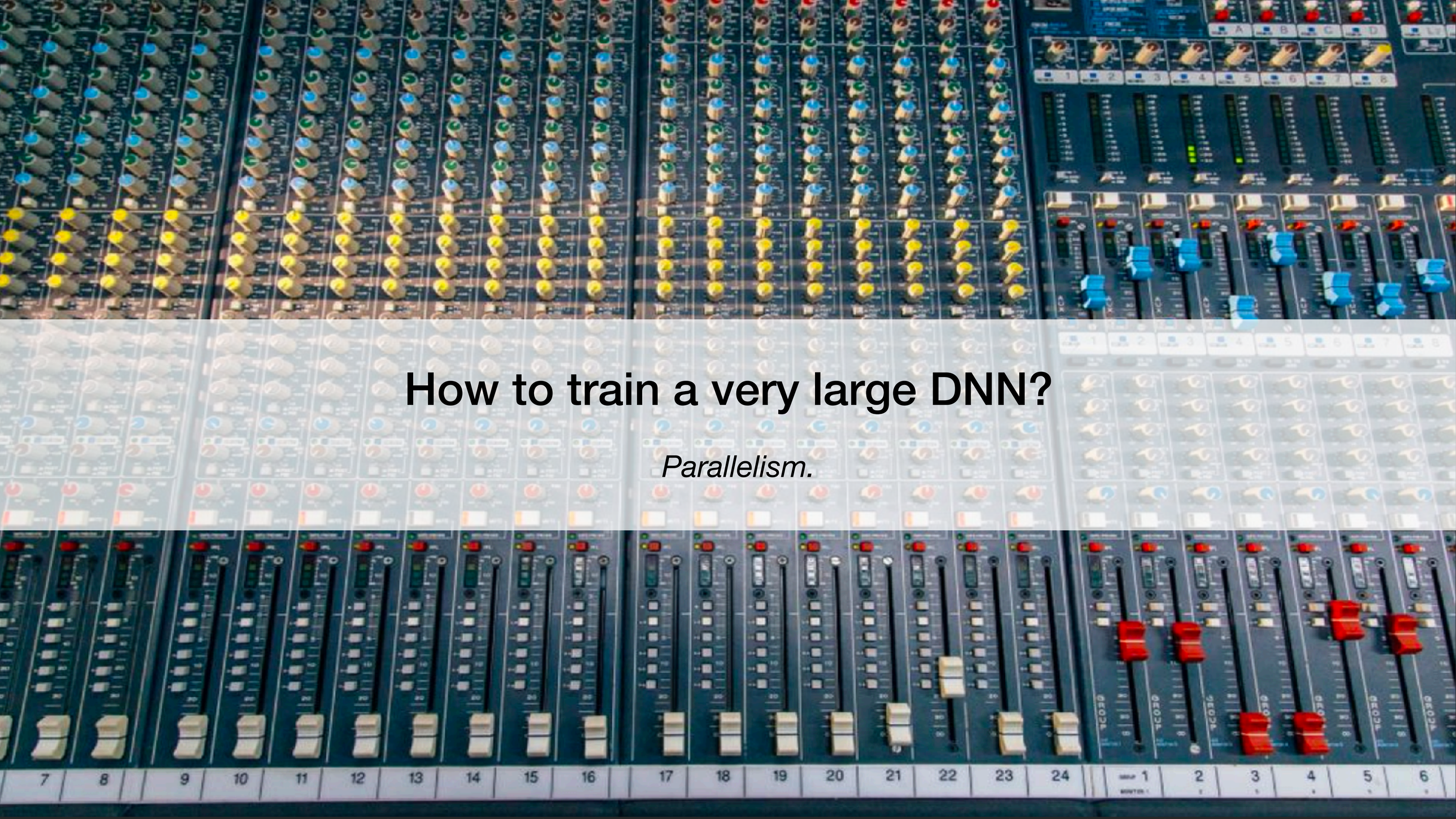
For today’s homework assignment, please describe the reasons for the US Civil War.

Output

By Donny Ferguson

It is easy to identify why the Civil War happened, because so many people and so many books and so much television and films tell us that it was the cause, that it has something to do with race or economics or religion. There is much agreement that it was essentially a war of slavery on behalf of capitalism, about a century of slavery. But that’s not what most people think of when they think of the Civil War. Many people think of the war as about states rights. You’re not wrong about that, but it’s kind of misleading to say that the Civil War was a conflict between states’ rights and federalism. So let’s try again. What’s the idea that has inspired so many people to go onto the battlefields of Gettysburg and Antietam and Gettysberg and Petersburg and Fredericksburg? The American idea of the republic--a notion of limited government--is a great part of the history.

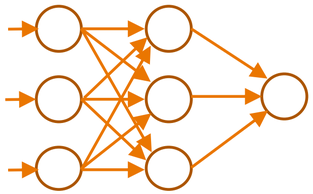
In 1791, Thomas Jefferson said “Our Constitution was made only for a moral and religious people. It is wholly inadequate to the government of any other.” The “founding fathers” were basically religious men who were not very



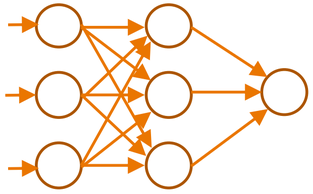
How to train a very large DNN?

Parallelism.

Data Parallel

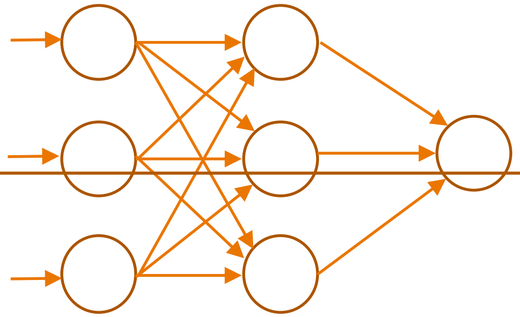


Processor 1



Processor 2

Model Parallel



Processor 1

Processor 2

Image credit: Caspar van Leeuwen

Learning: Optimizing the loss function

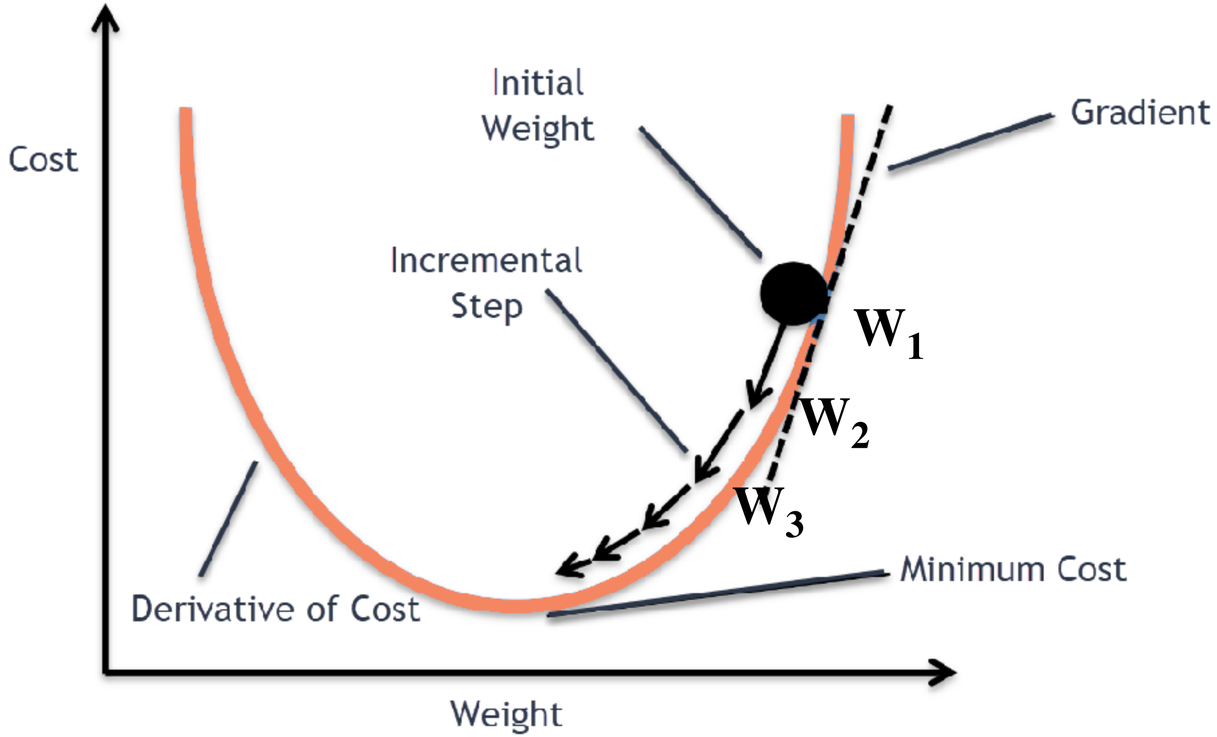
Objective: minimize the **difference** between predicted value the actual value

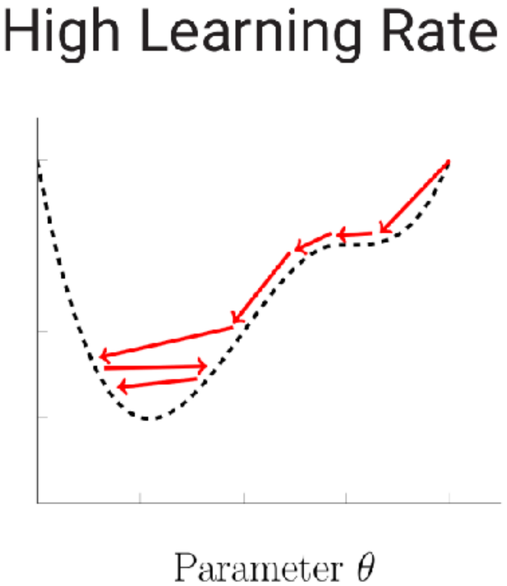
$$\mathscr{L}(\mathbf{W}, b) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2$$

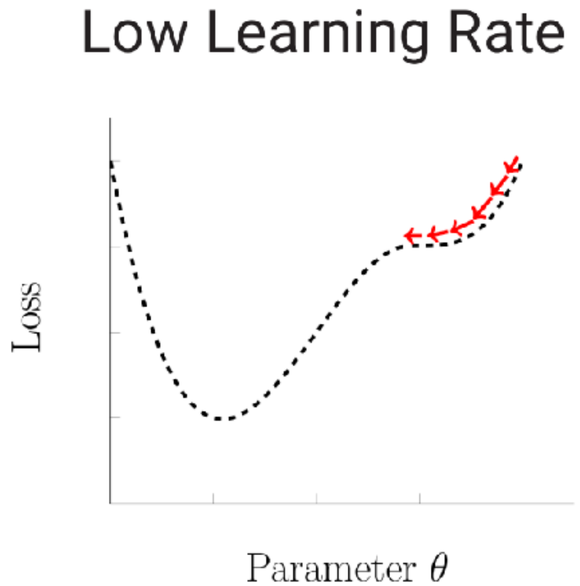
$$\nabla\mathscr{L}(\mathbf{W}, b)$$

$$\mathbf{W}_{j+1} = W_j - \alpha\nabla\mathscr{L}(\mathbf{W}_j, b)$$

Learning rate







Parallel Gradient Descent

Single processor

$$\mathscr{L}(\mathbf{W}, b) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \qquad \mathbf{W}_{j+1} = W_j - \alpha\nabla\mathscr{L}(\mathbf{W}_j, b)$$

Multiple processors

$$g_1 = \nabla\mathscr{L}(\mathbf{W}_j, b)$$

$$g_2 = \nabla\mathscr{L}(\mathbf{W}_j, b)$$

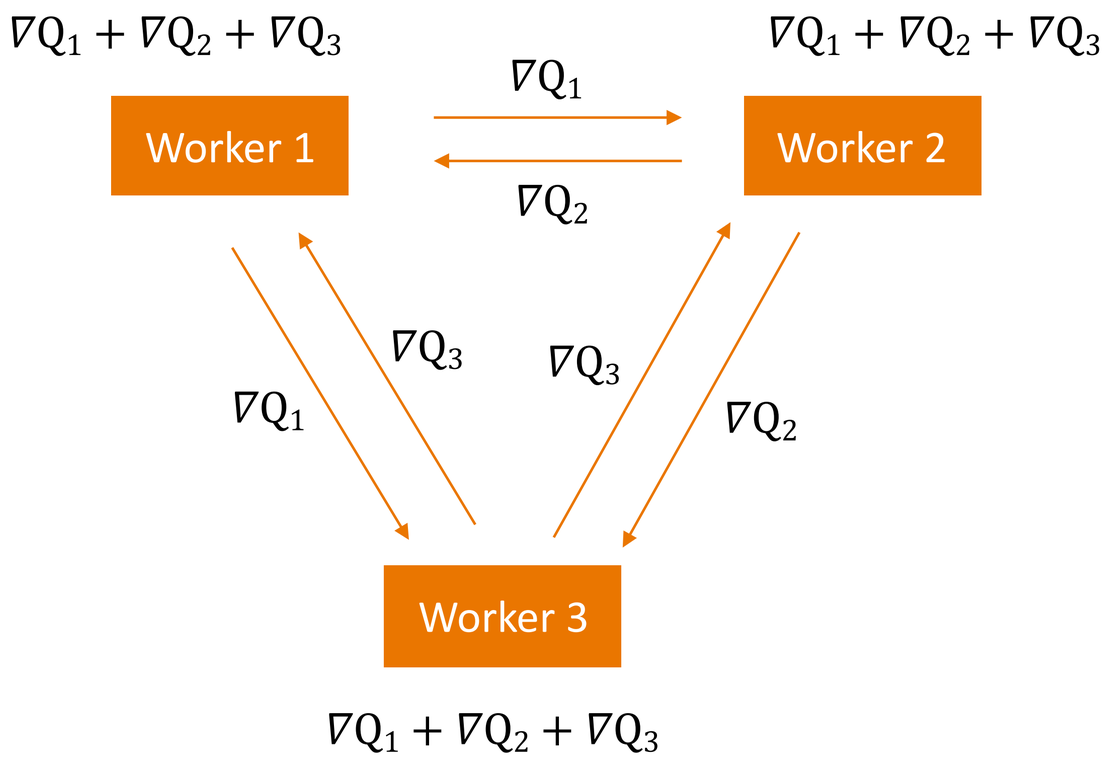
$$g_3 = \nabla\mathscr{L}(\mathbf{W}_j, b)$$

$$g_4 = \nabla\mathscr{L}(\mathbf{W}_j, b)$$

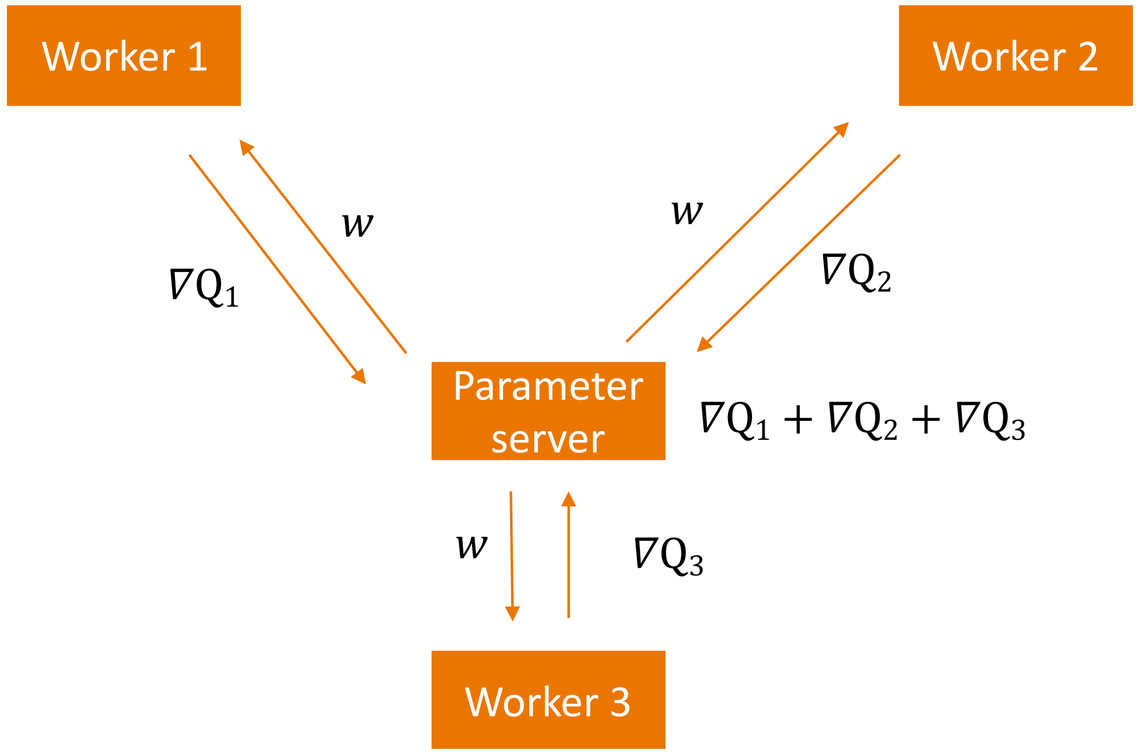
$$\nabla\mathbf{Q} = \Sigma_1^N g_n$$

$$\mathbf{W}_{j+1} = \mathbf{W}j - \alpha\nabla\mathbf{Q}$$

Gradient Reduction Topology (sync'ed)



Decentralized



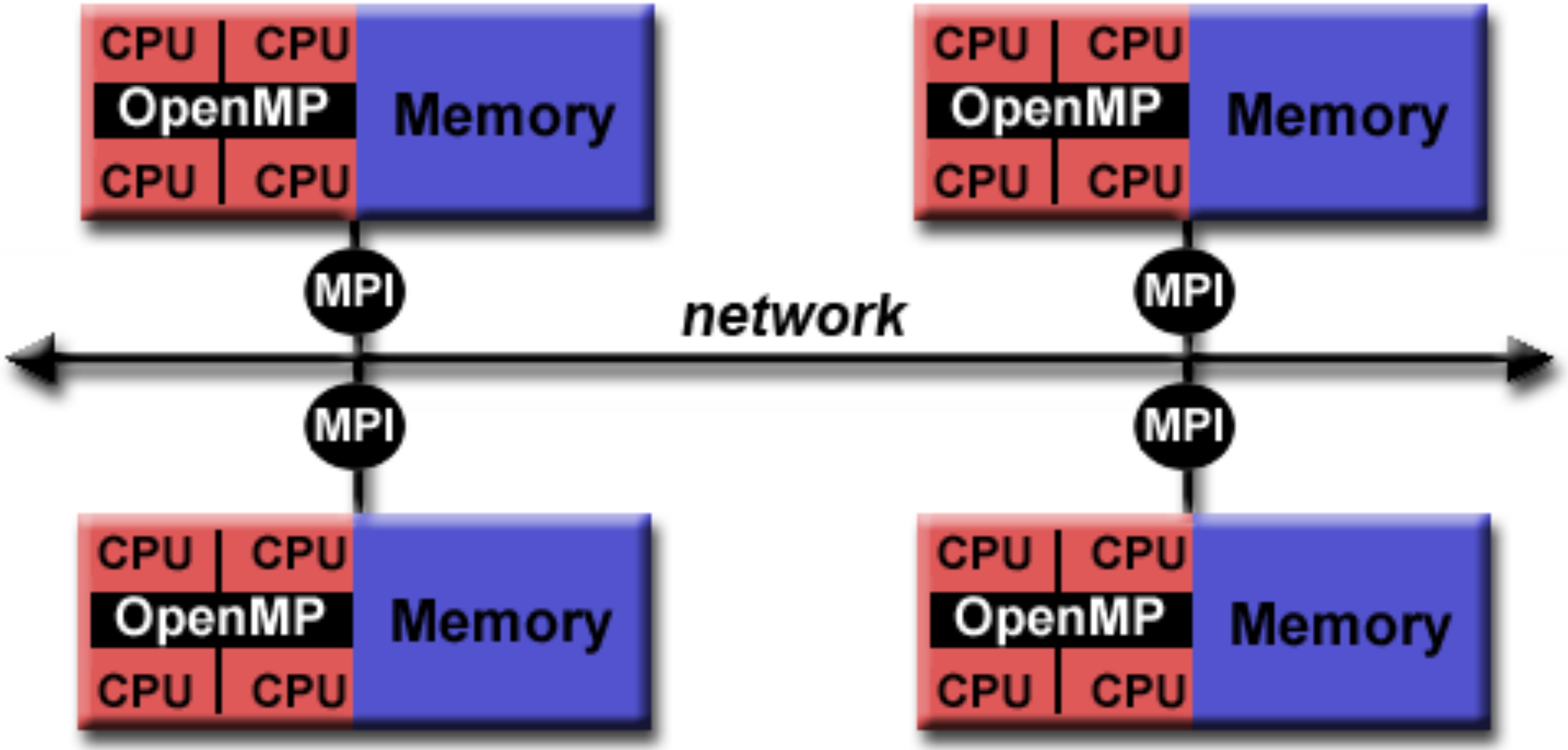
Centralized

More info, see e.g., https://arxiv.org/pdf/1705.09056.pdf

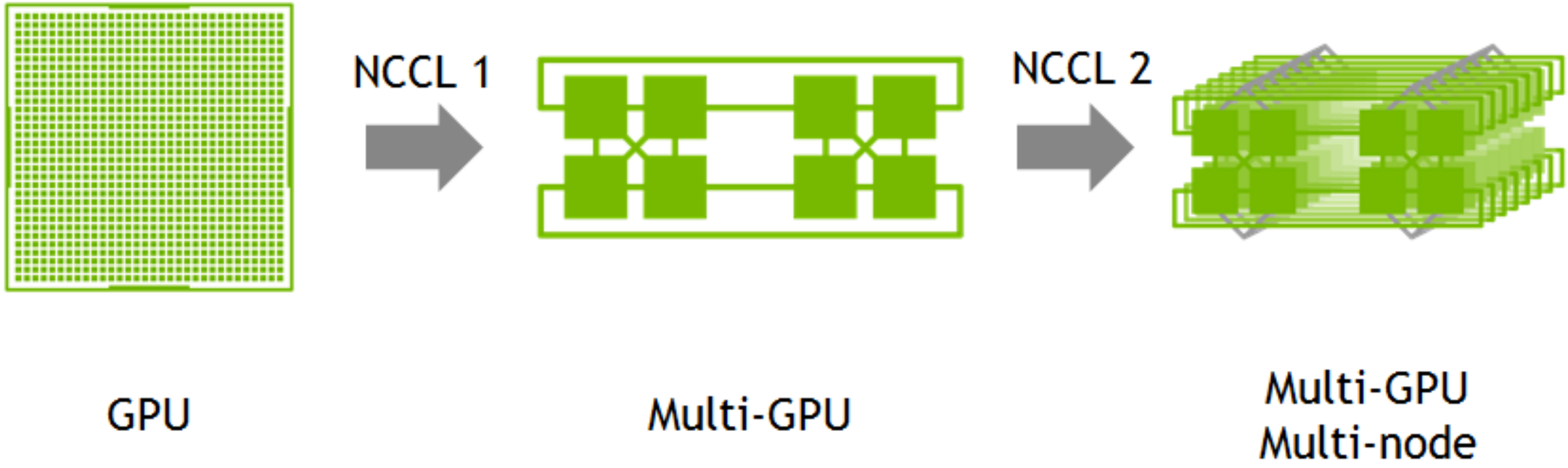
Image credit: Caspar van Leeuwen

How do we send gradients?

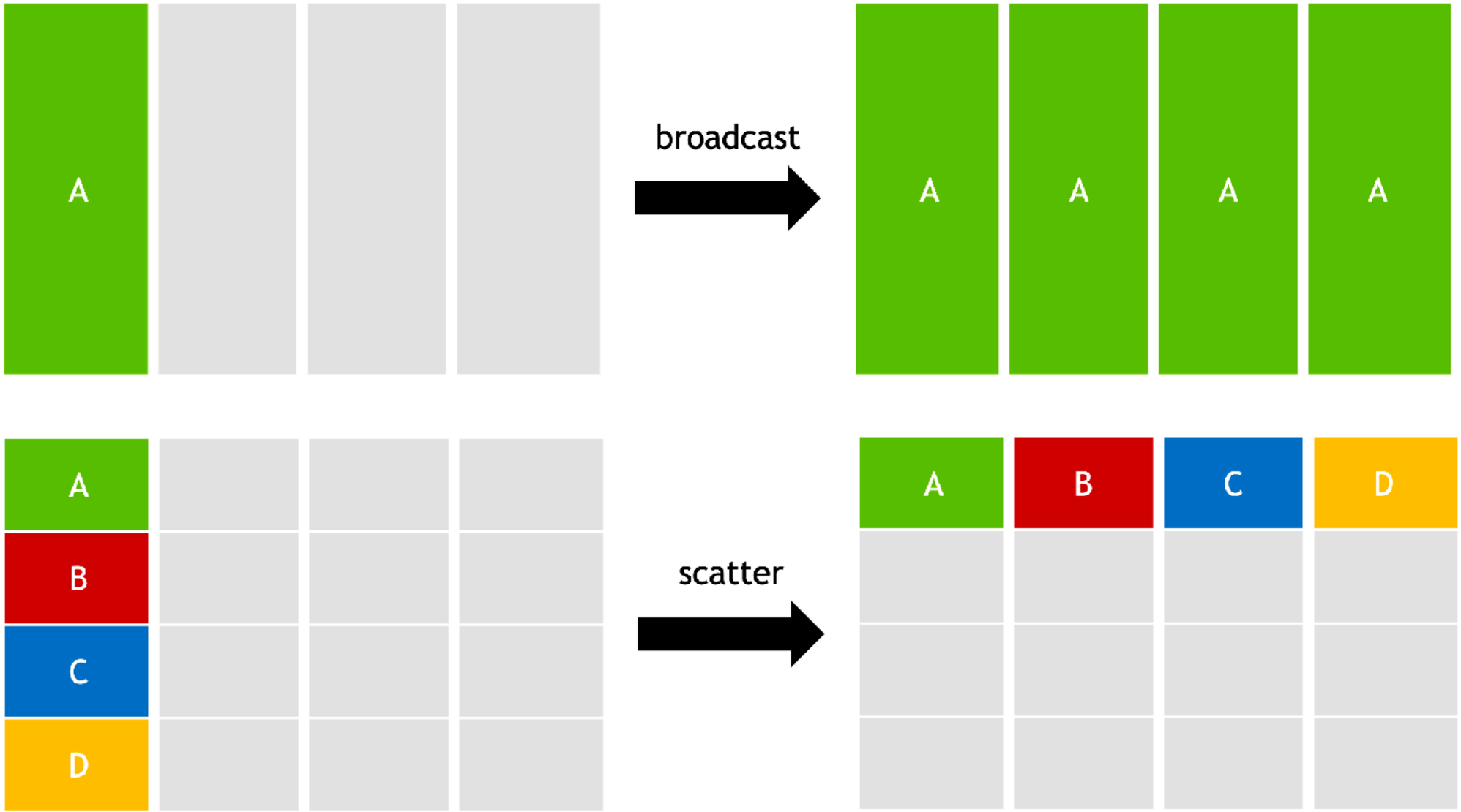
Message Passing Interface (MPI)



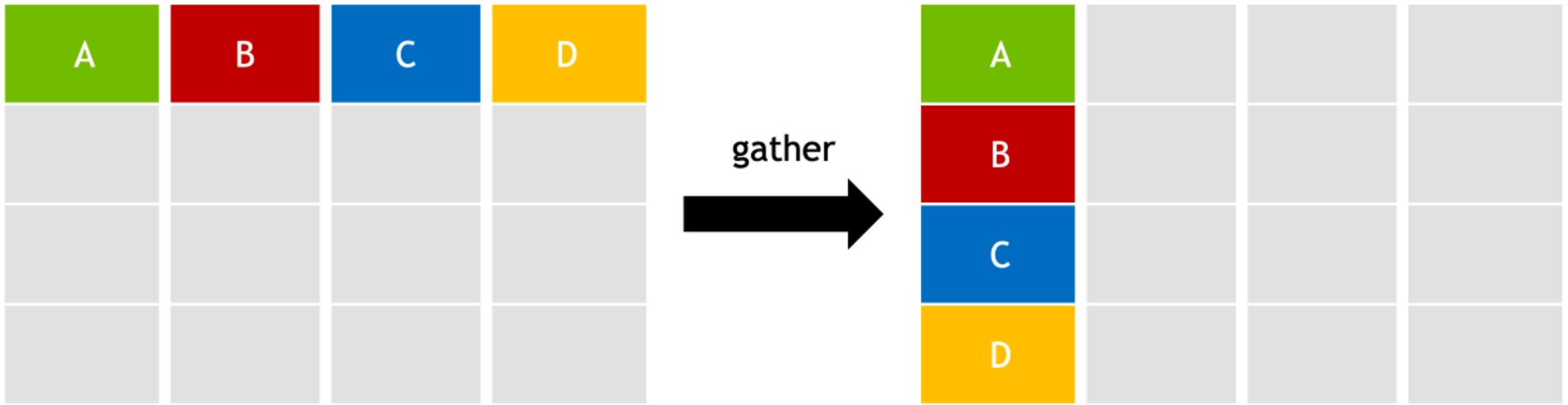
NVIDIA Collective Communication Library (NCCL)

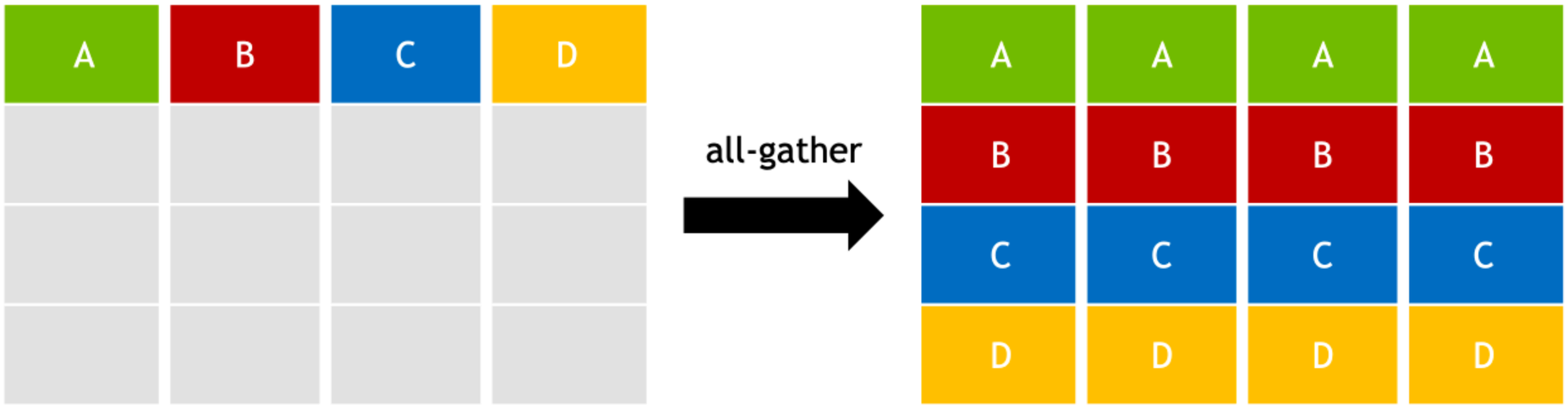


Collective Communication

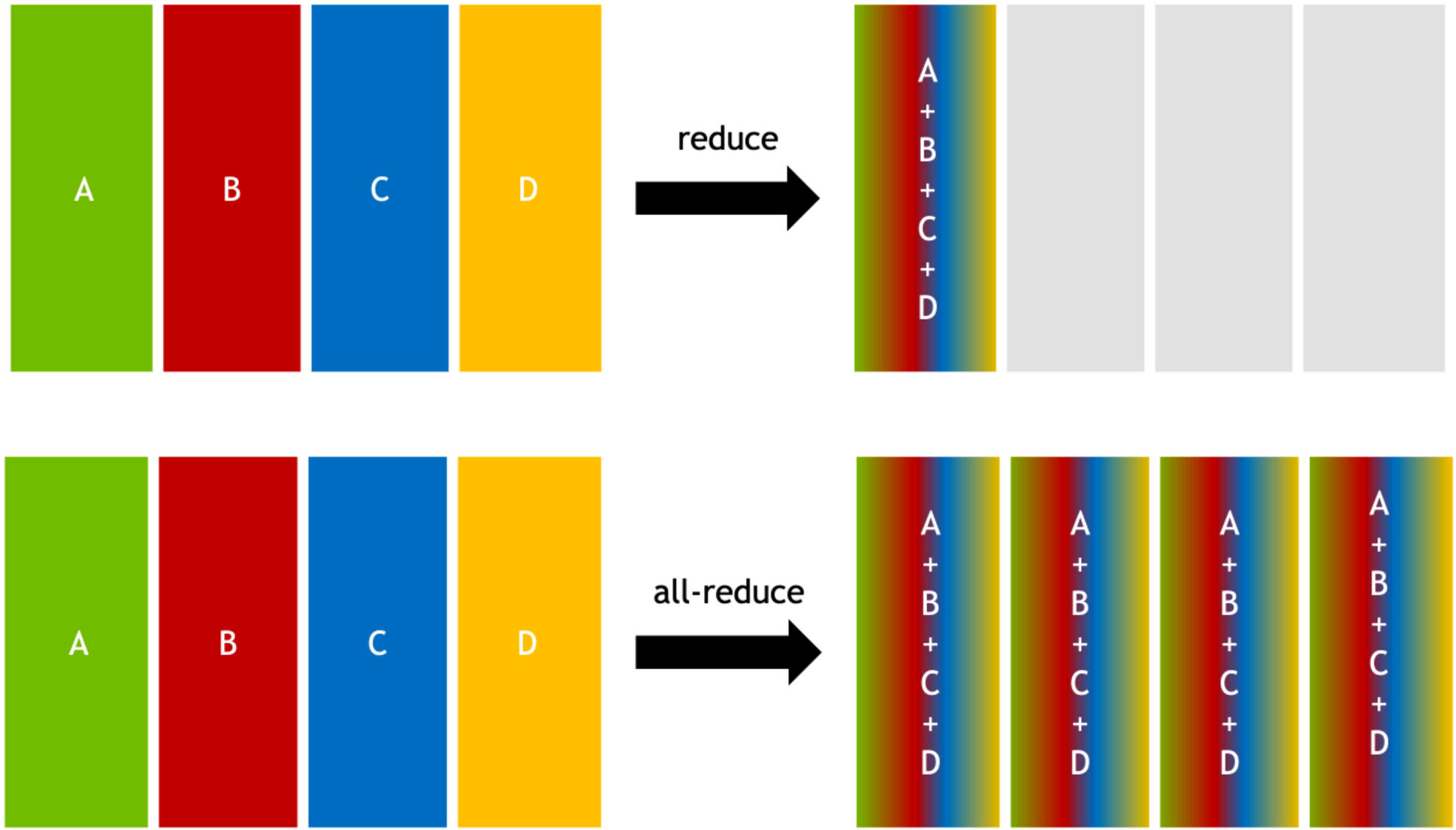


Collective Communication

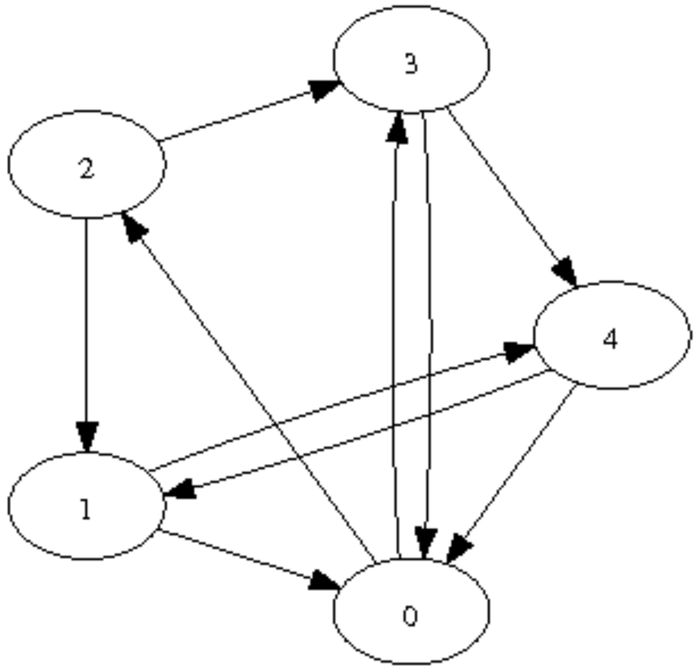




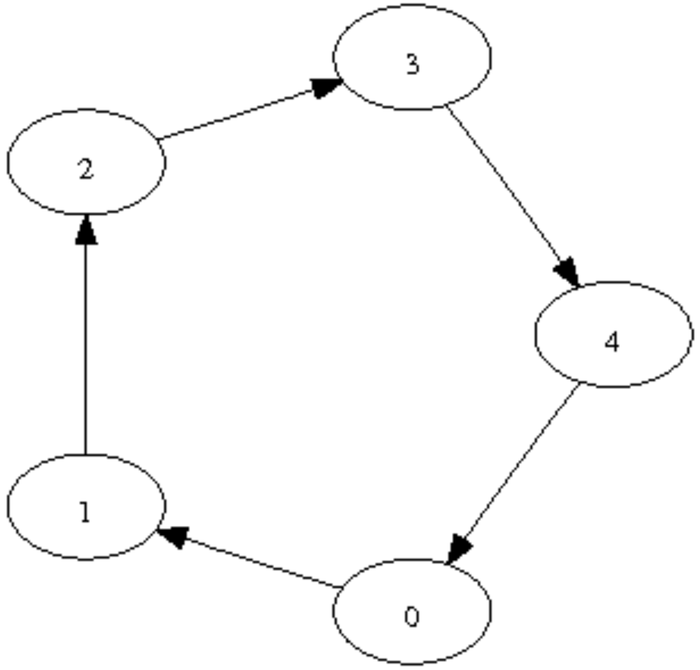
Collective Communication



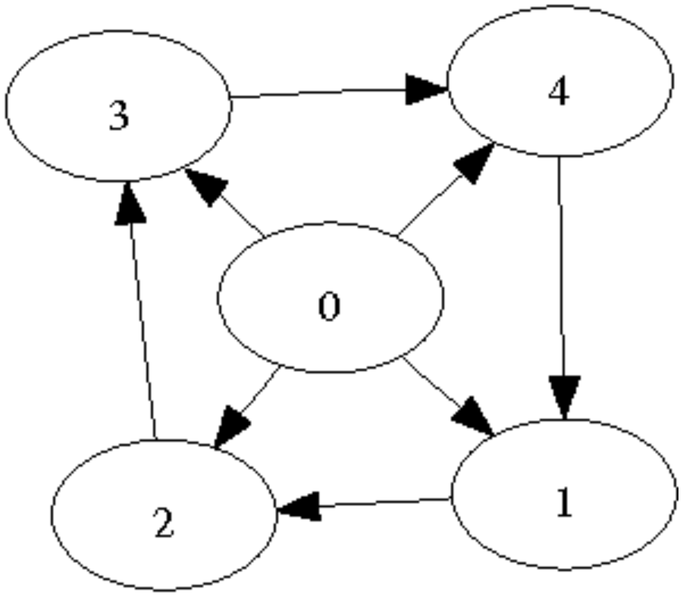
Communication Topology



(a) Random



(b) Ring



(c) Wheel

DL Framework Support

TensorFlow

CPU, GPU, TPU
Multi-GPU on one node with `tf.device`
Multi-node support with `tf.distribute`
`cuDNN, MKL-DNN`
`MPI, NCCL`

PyTorch

CPU, GPU
Multi-GPU on one node with `torch.multiprocessing`
Multi-node support with `torch.distributed`
`cuDNN, MKL-DNN`
`Gloo, MPI, NCCL`

Or use a framework on the top of DL libraries

- Supports TensorFlow, Keras, PyTorch, MXNet
- Supports MPI and NCCL
- Just a few lines of code changes (in principle)
- Has its own profiling facility (Horovod Timeline)