

NetTorrent DSC-PESU Projects

Team Members:

Sparsh Temani

Madhav Jivrajani

Problem Statement

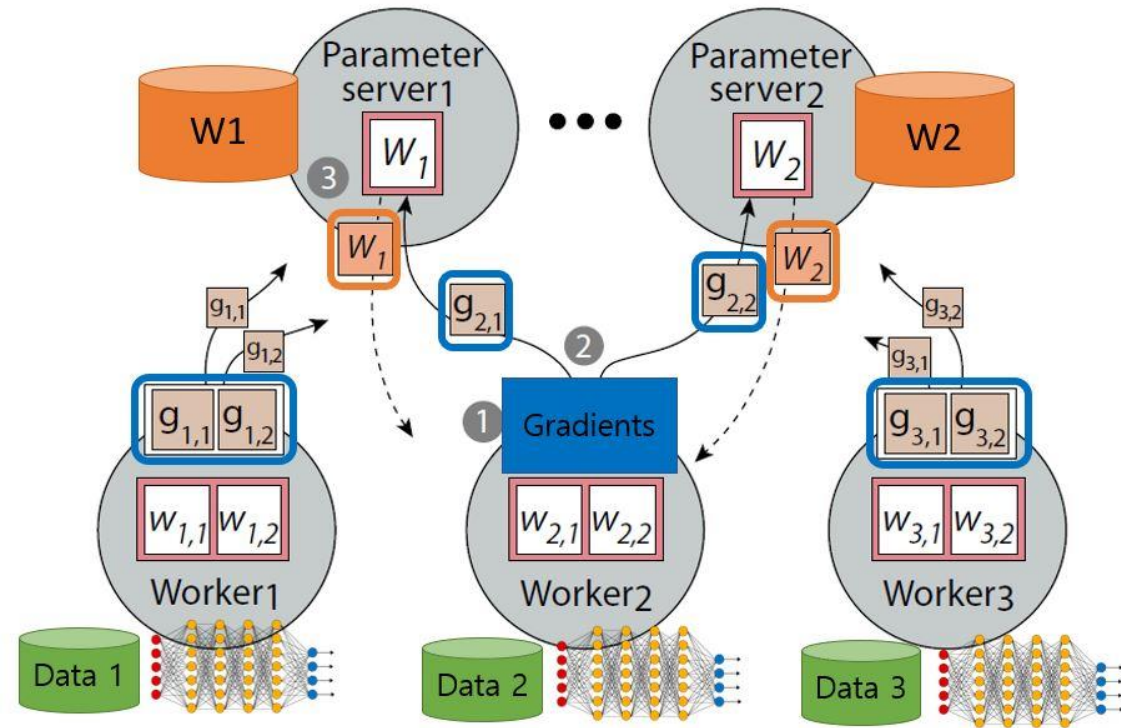
"To build a decentralized and distributed protocol for training Deep Neural Networks in order to achieve a speed-up in training time with almost no loss in accuracy as compared to centralized and traditional distributed models."

Why do we need this?

- The amount of data being generated every single day increases exponentially.
- Deep Neural Networks have revolutionized the accuracy machine learning models can achieve.
- There is an intrinsic need to scale DNN training in a horizontal manner.
- As a result, scalable distributed systems that train DNNs parallelly on clusters and can manage large volumes of training data in a manageable amount of time were introduced.

What's
already out
there?

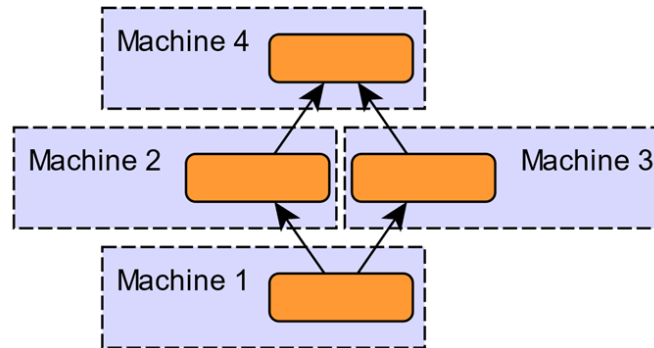
- The goal of current systems is to perform training of DNNs in a distributed manner.
- Most of these systems follow a master-slave architecture with a parameter server which is responsible for synchronizing replica models training at the slave nodes.



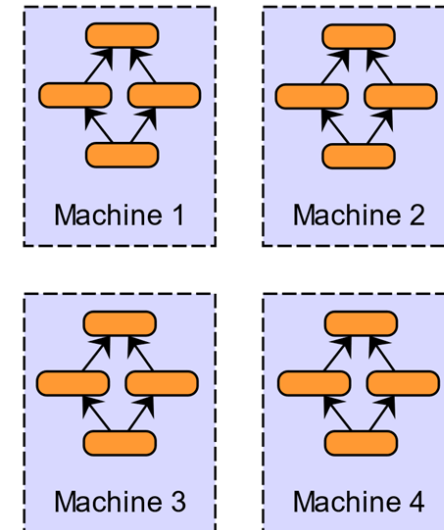
What's
already out
there?

Data and Model Parallelism

Model Parallelism



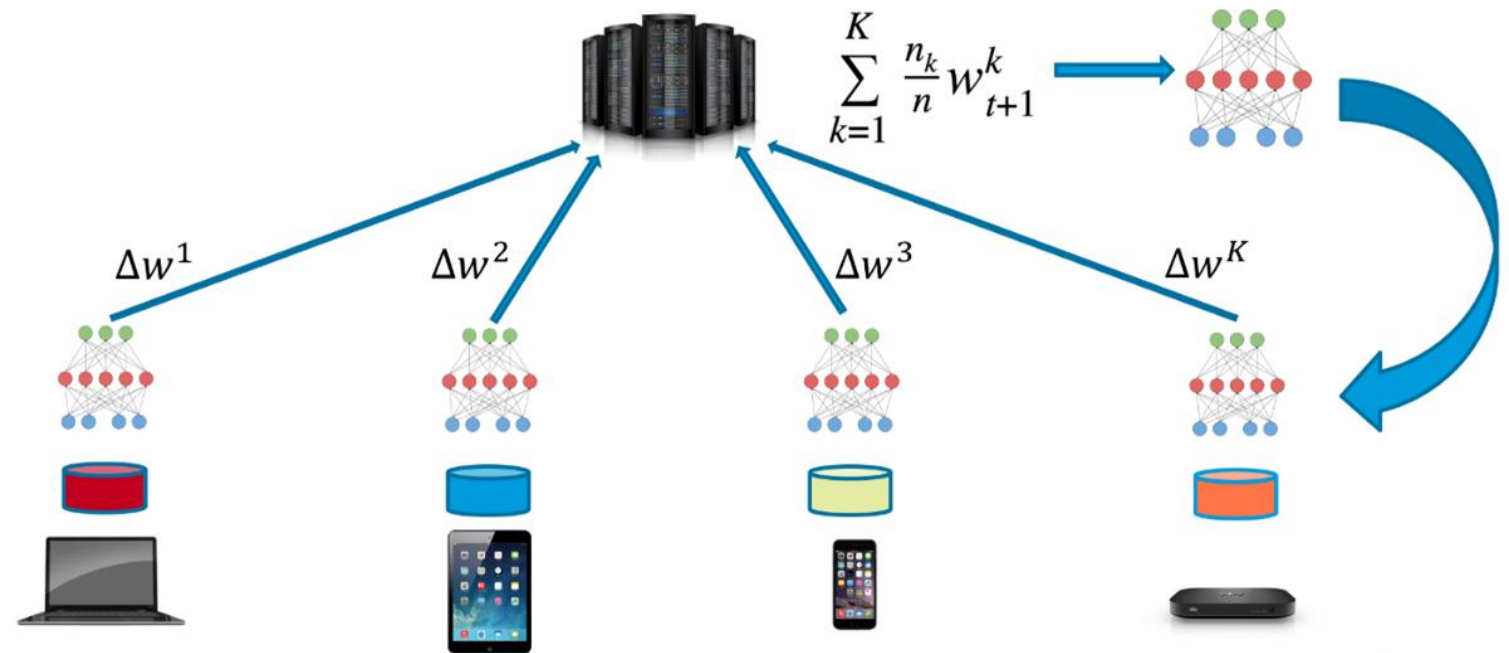
Data Parallelism



What's
already out
there?

Federated Learning

Large scale data-parallelism

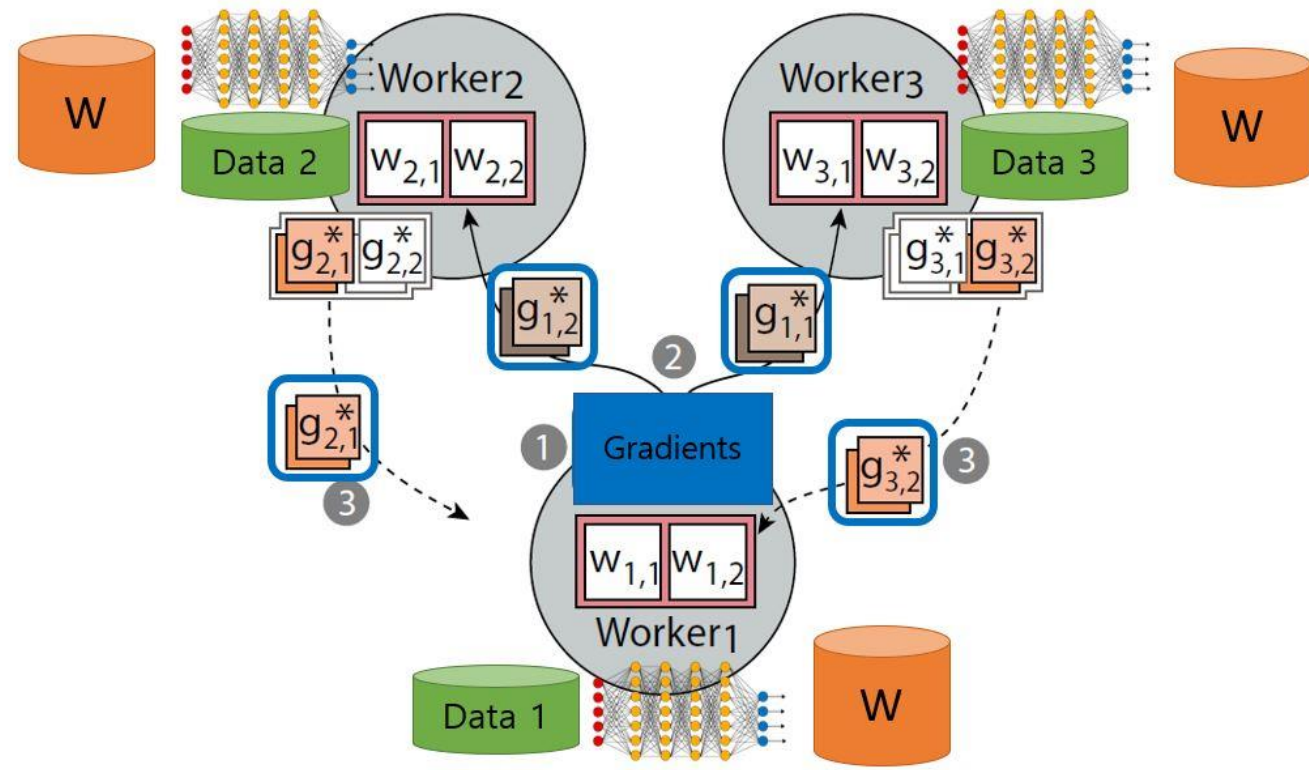


Drawbacks of current distributed frameworks

- Tradeoff between computational resources and network resources to achieve the fastest model convergence.
- Division of resources is dependent on a lot of factors.
- Optimal division of resources is done by trial and error, highly cumbersome.
- Parameter server is a single point of failure.

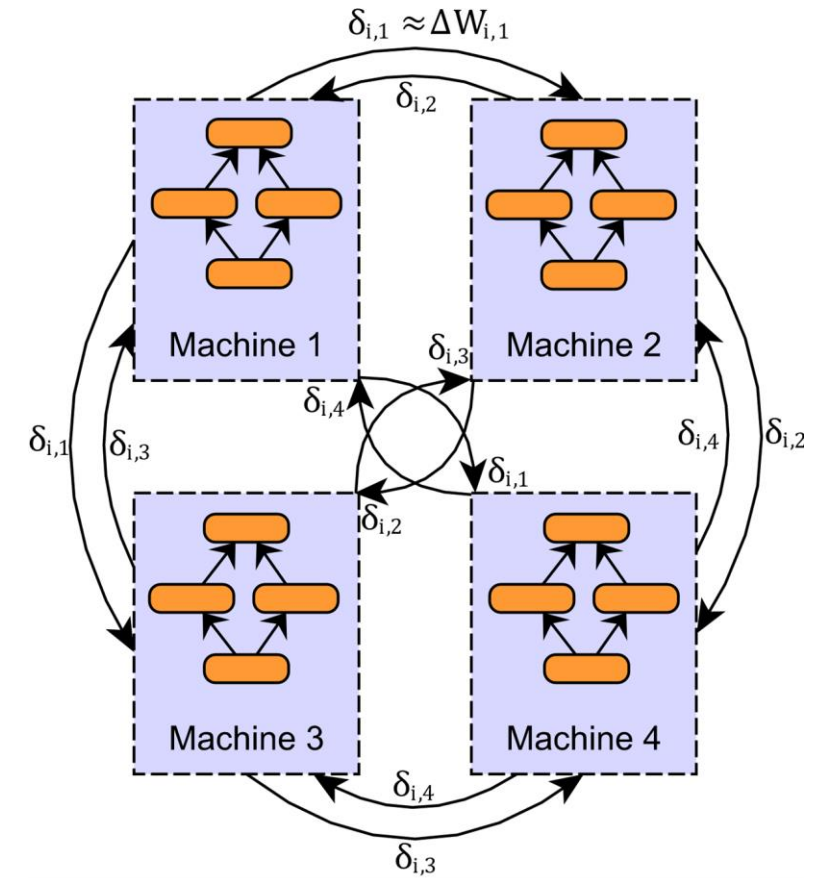
NetTorrent

*We propose a decentralized and distributed system which does not use parameter servers but only homogenous workers employing data **and** model parallelism.*



NetTorrent

- Asynchronous model updation is done in a purely peer-to-peer manner.
(Data Parallelism)
- Computation is independent of network bandwidth consumption.
- Partial Gradient Exchange.

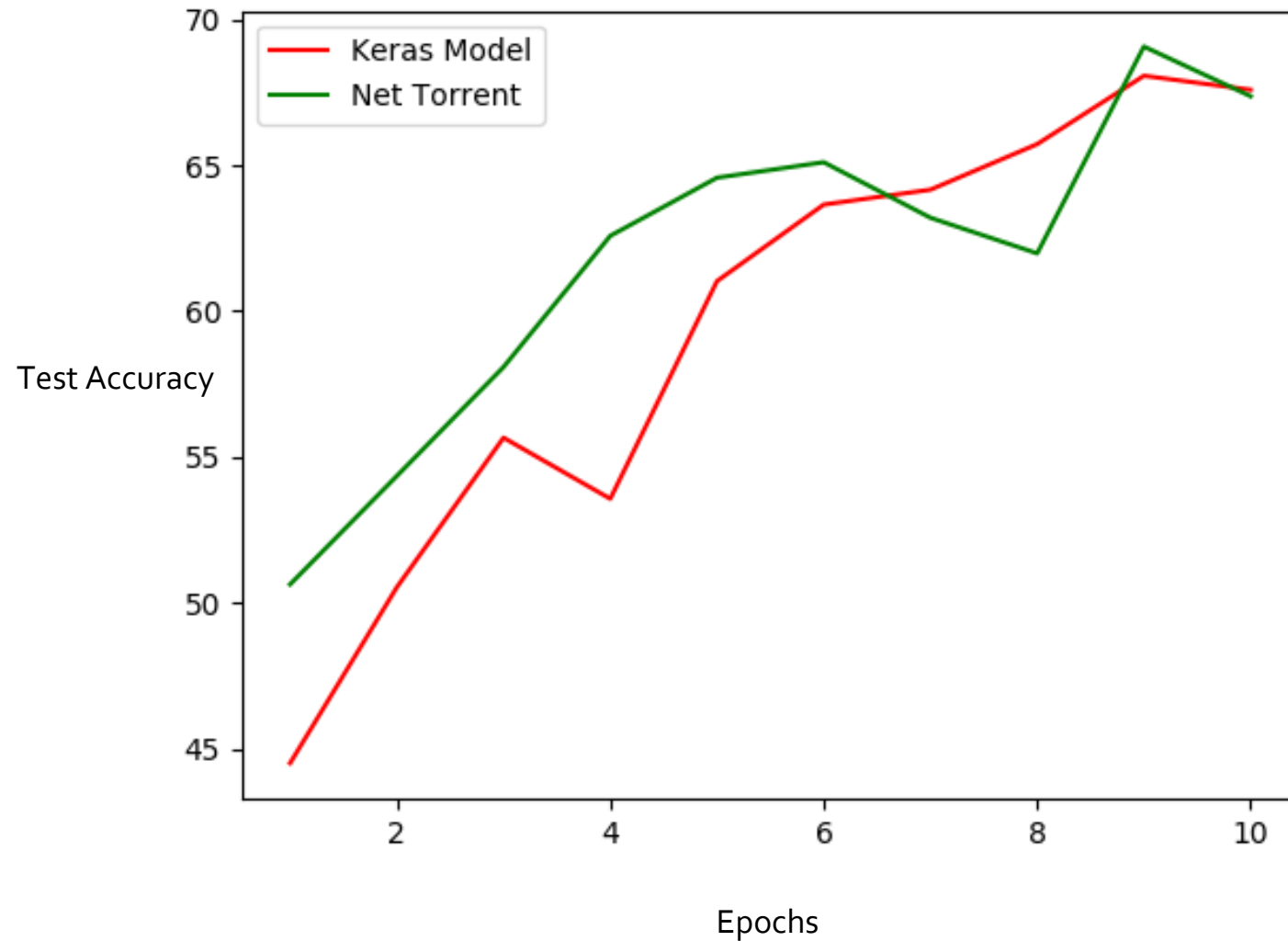


Results

Time Comparison	Normal Training Time	NetTorrent
MNIST	601 seconds	362 seconds
CIFAR - 10	251 seconds	115 seconds

Results

Comparison of performance : Test Accuracy Vs Epochs (CIFAR-10)



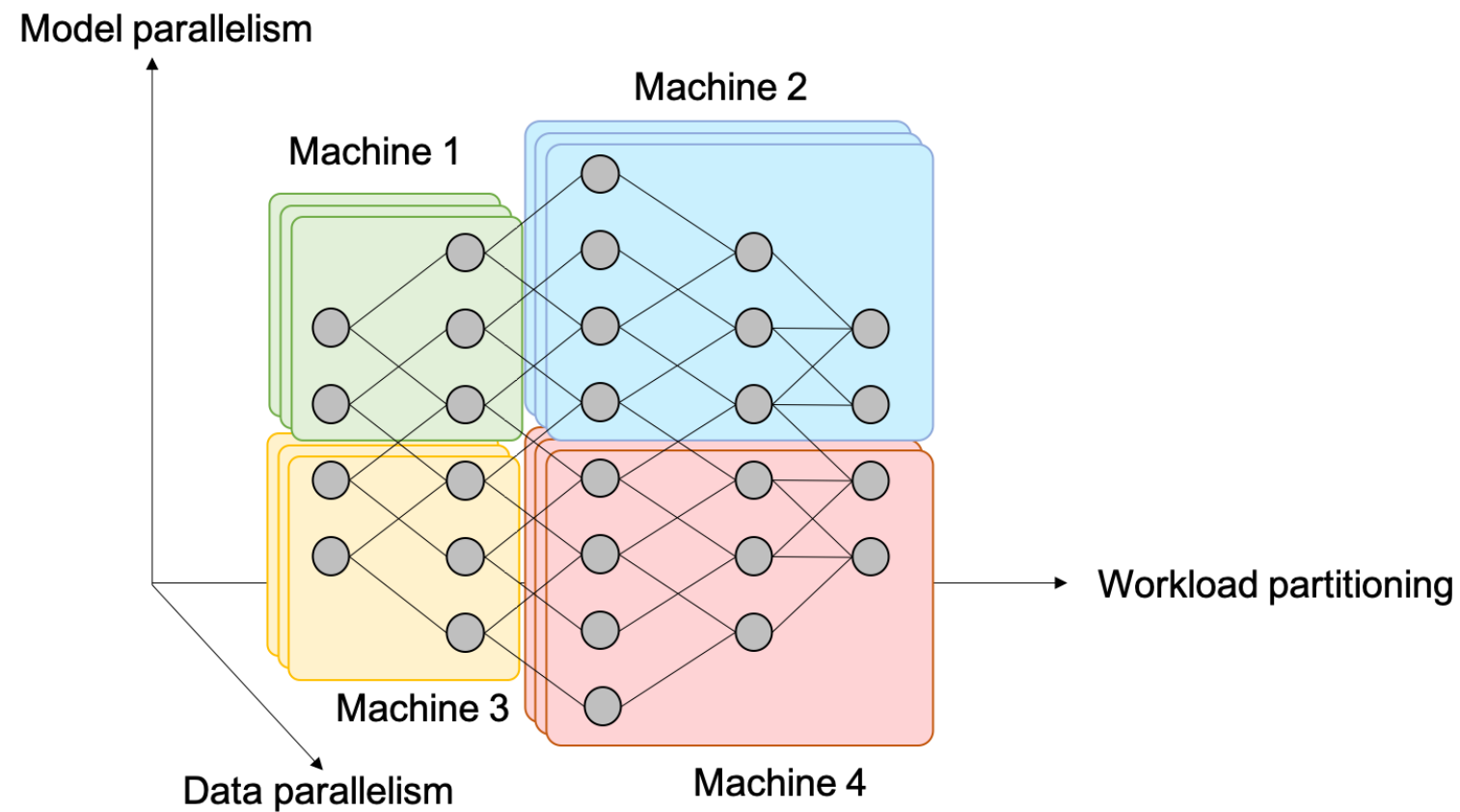
NetTorrent: Current Progress

- Implemented a basic and unstructured p2p network.
- Successfully partitioned data among nodes.
- Built a proof of concept by achieving data parallelism between 2 nodes.

NetTorrent: Road Map

- The next step for us will be to build a scalable decentralized system which employs data parallelism.
- Once that is done, the next step would be to implement model parallelism as well.
- We do this by having a separate p2p network for training individual layers and hope to achieve more speed-up in training time.
- Backpropagation would be achieved by applying asynchronous optimization algorithms such as ASGD.

NetTorrent: Road Map



Thank you!