Exploring the Use of Synthetic Gradients for Distributed Deep Learning across Cloud and Edge Resources

Yitao Chen, Kaiqi Zhao, Baoxin Li, Ming Zhao

Arizona State University

http://visa.lab.asu.edu

Background





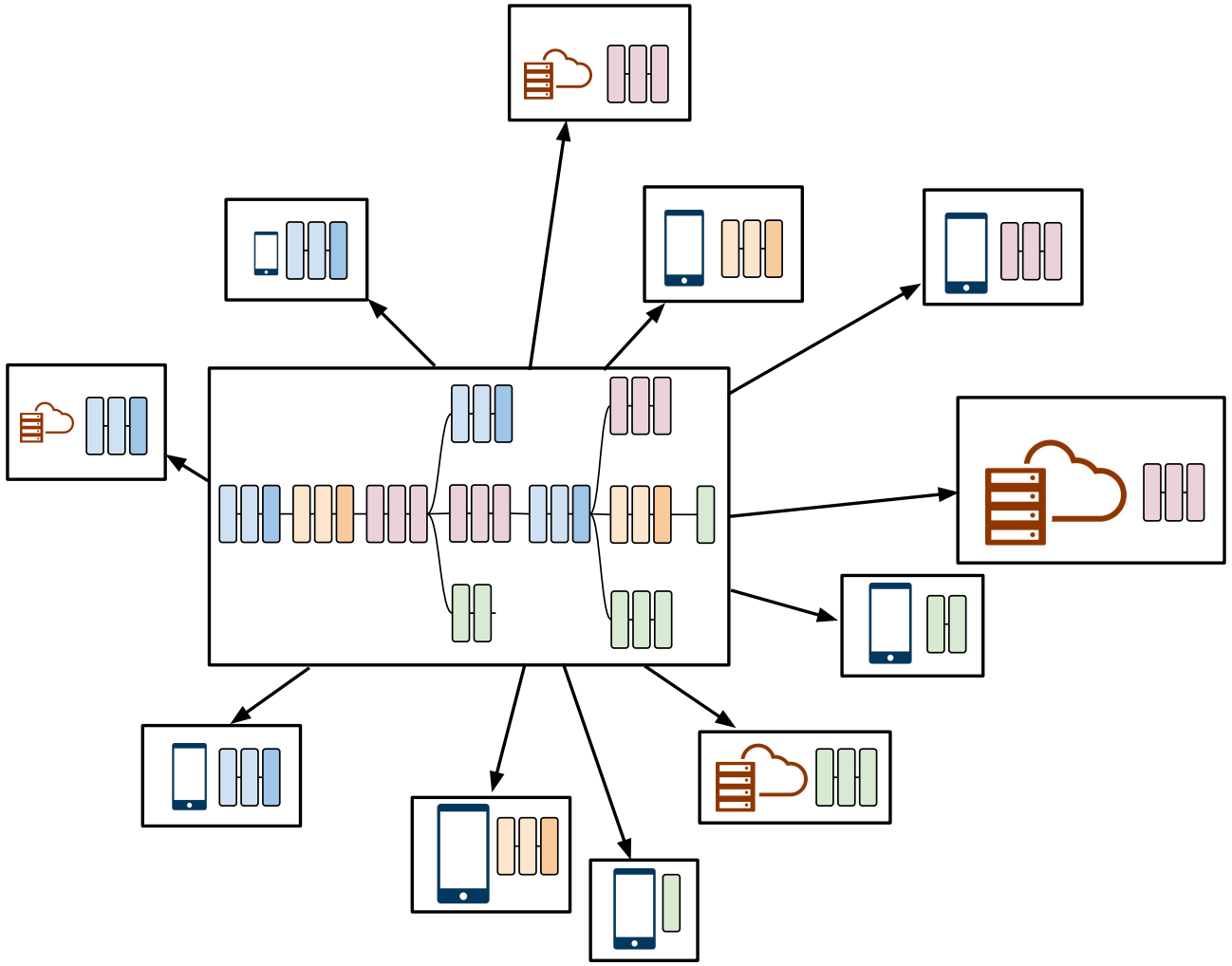
NOT SUFFICIENT!

- ~~Cloud-only approach: training models only in the cloud~~

Limitations of cloud-only approach

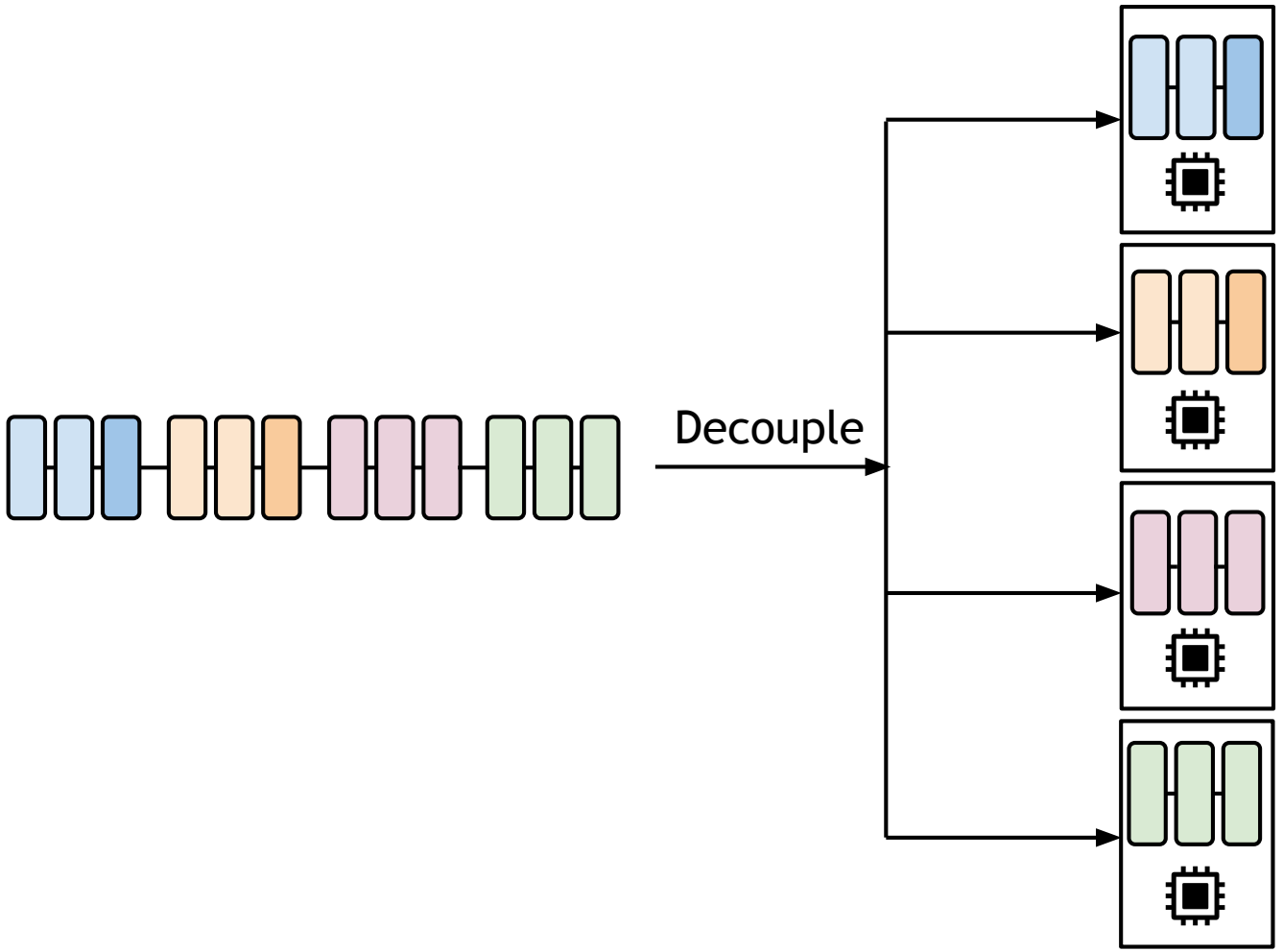
- Cloud is insufficient
  - Not enough resources when facing explosive growth of data
  - No fast enough to provide responsive training on real-time data
- Unable to exploit the increasingly powerful edge devices in training
  - Multi-core, GPUs, AI chips
  - Edge devices are not contributing to the deep learning training

Objectives



- Distributed learning across cloud and edge resources
  - Utilize potentially weak resources
  - Across edge devices and cloud resources
- Limitations of existing solutions
  - Data parallelism does not work for edge resources
  - Conventional backpropagation based training makes model parallelism difficult

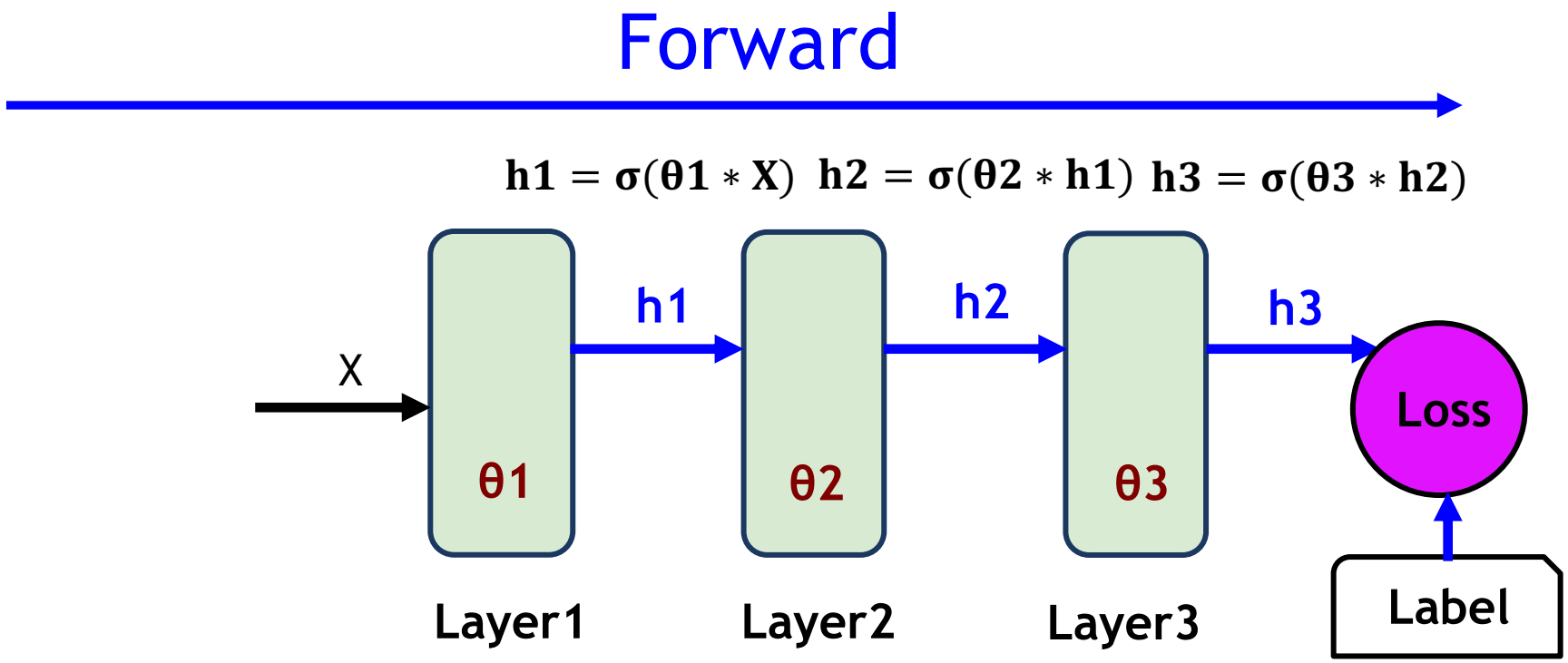
Proposed approach



Using synthetic gradients to achieve model parallelism

- Split a large DNN model into many small models
- Deploy small models on distributed and potentially weak resources
- Use synthetic gradients to decouple the training of the small models

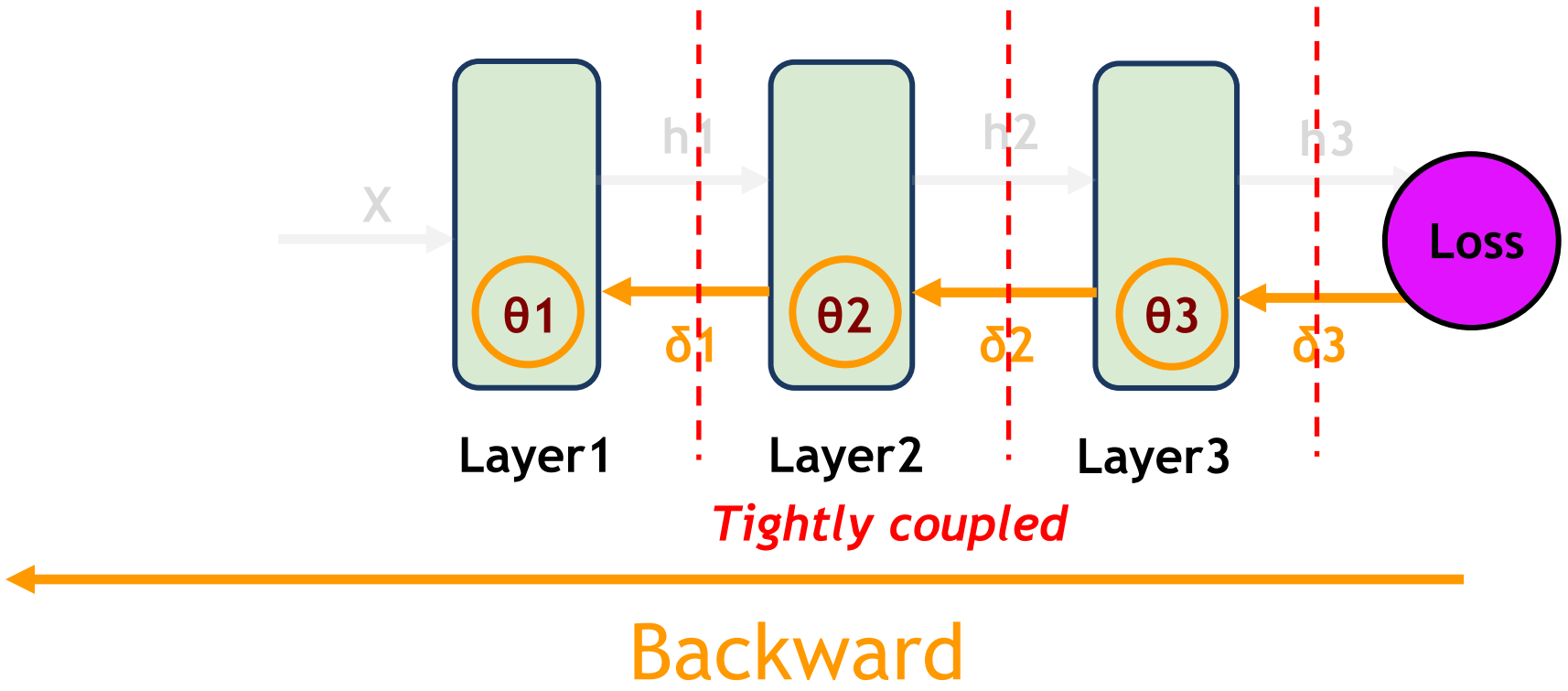
Conventional backpropagation training



Conventional backpropagation training

Decouple the layers in backpropagation by predicting the gradients!

Backpropagation tightly coupling the layers!



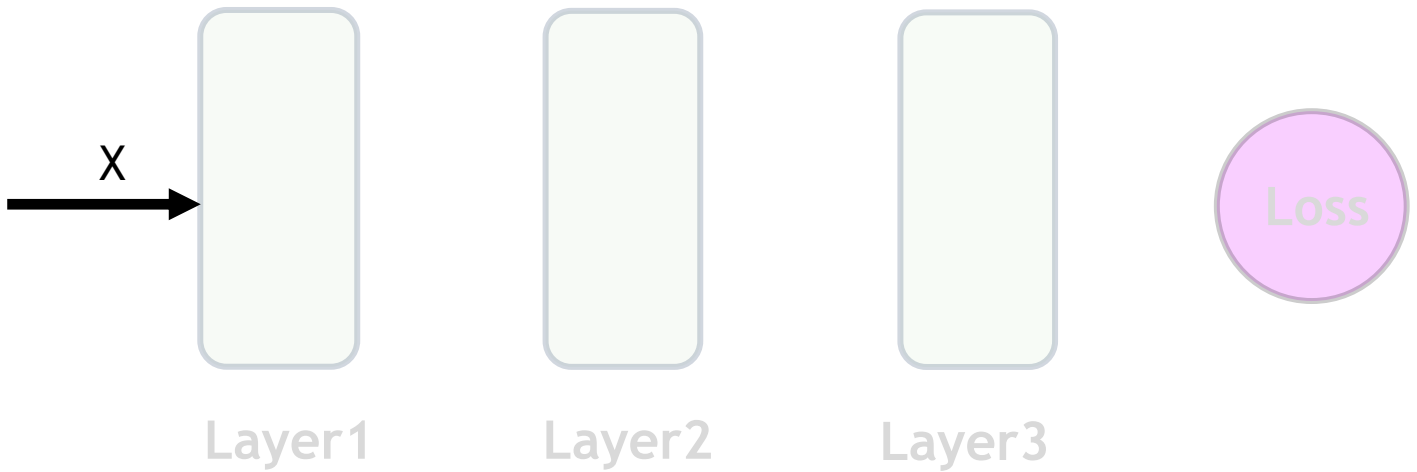
Loss = L

$$\frac{dL}{d\theta 3} = \frac{dL}{dh3} * \frac{dh3}{d\theta 3}$$

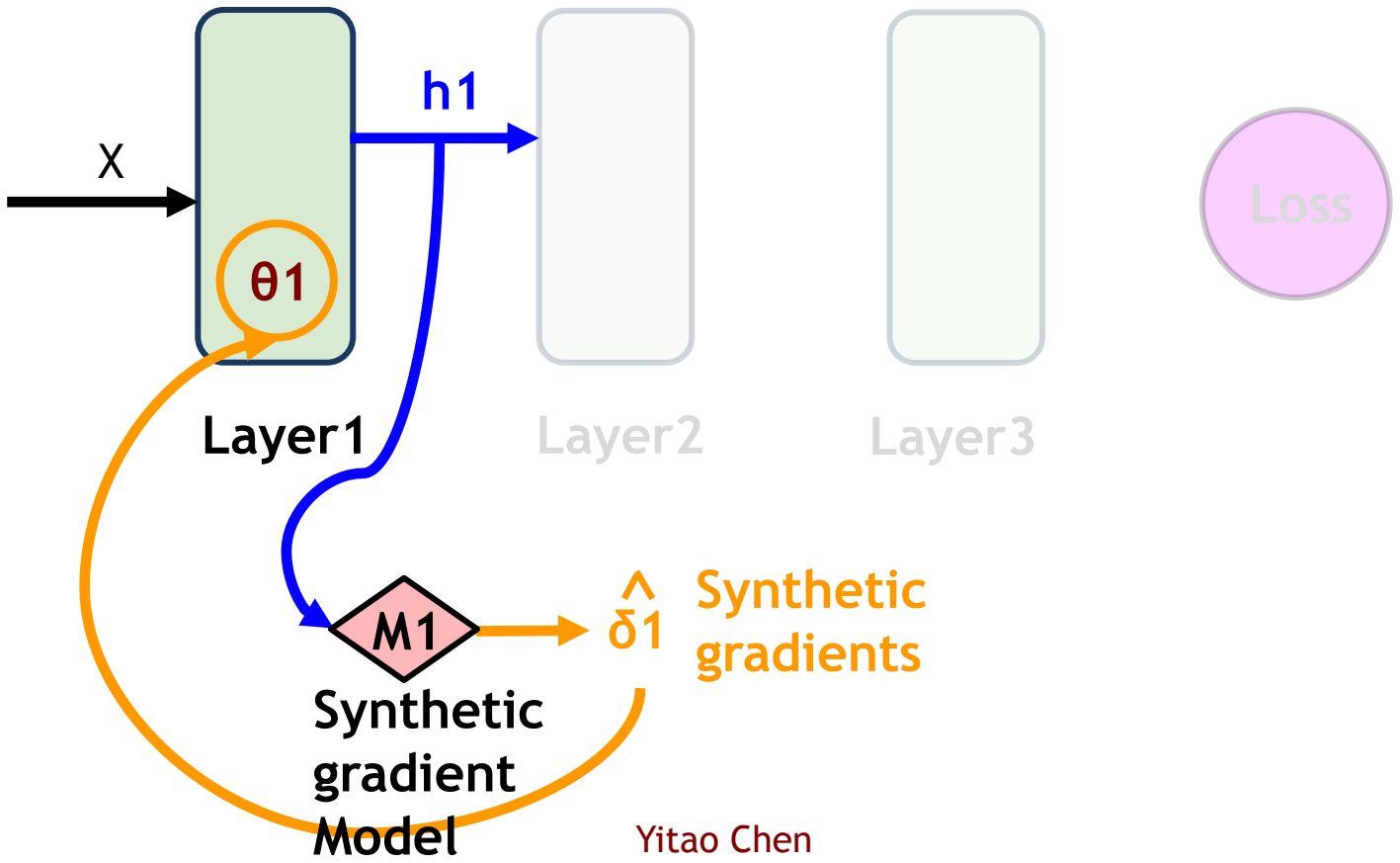
$$\frac{dL}{d\theta 2} = \frac{dL}{dh2} * \frac{dh2}{d\theta 2}$$

$$\delta 2 = \frac{dL}{dh3} * \frac{dh3}{dh2}$$

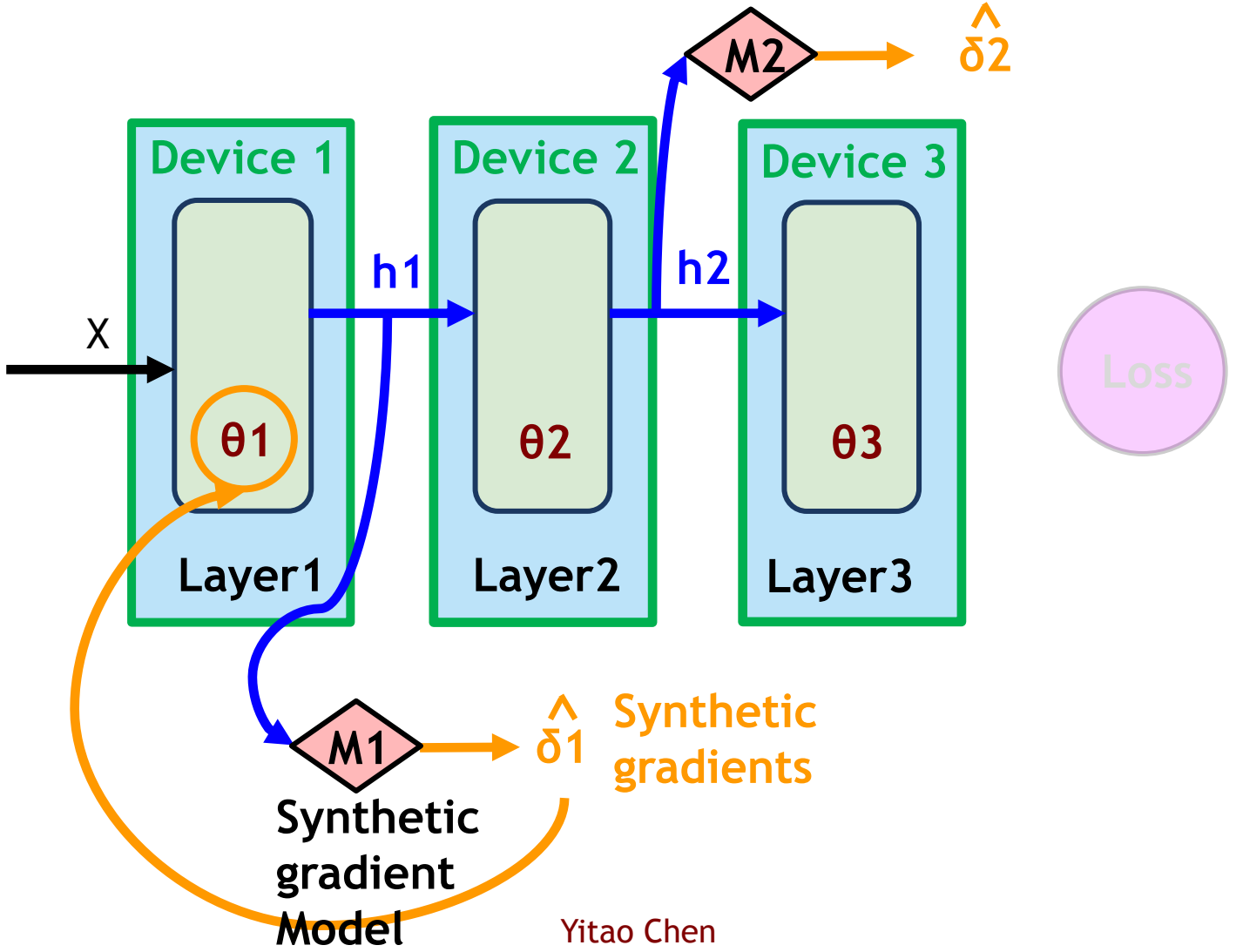
Decouple layers using synthetic gradients



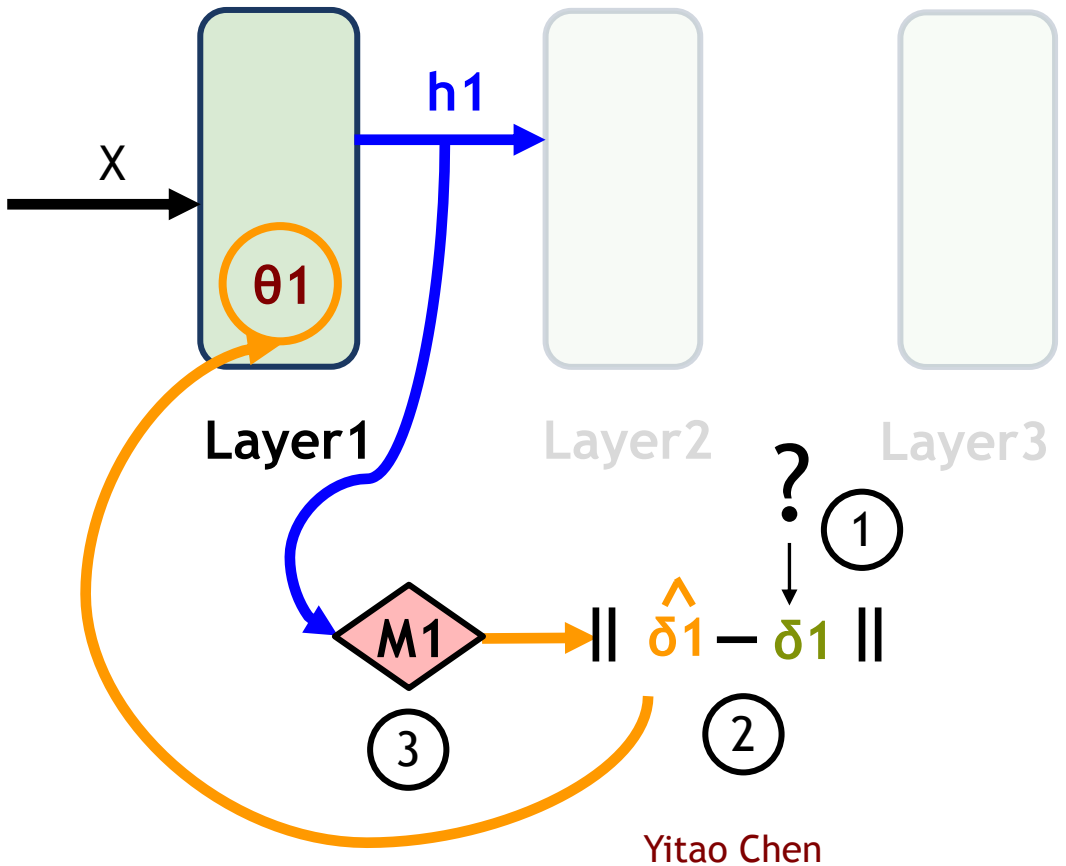
Decouple layers using synthetic gradients



Decouple layers using synthetic gradients



Gradient prediction





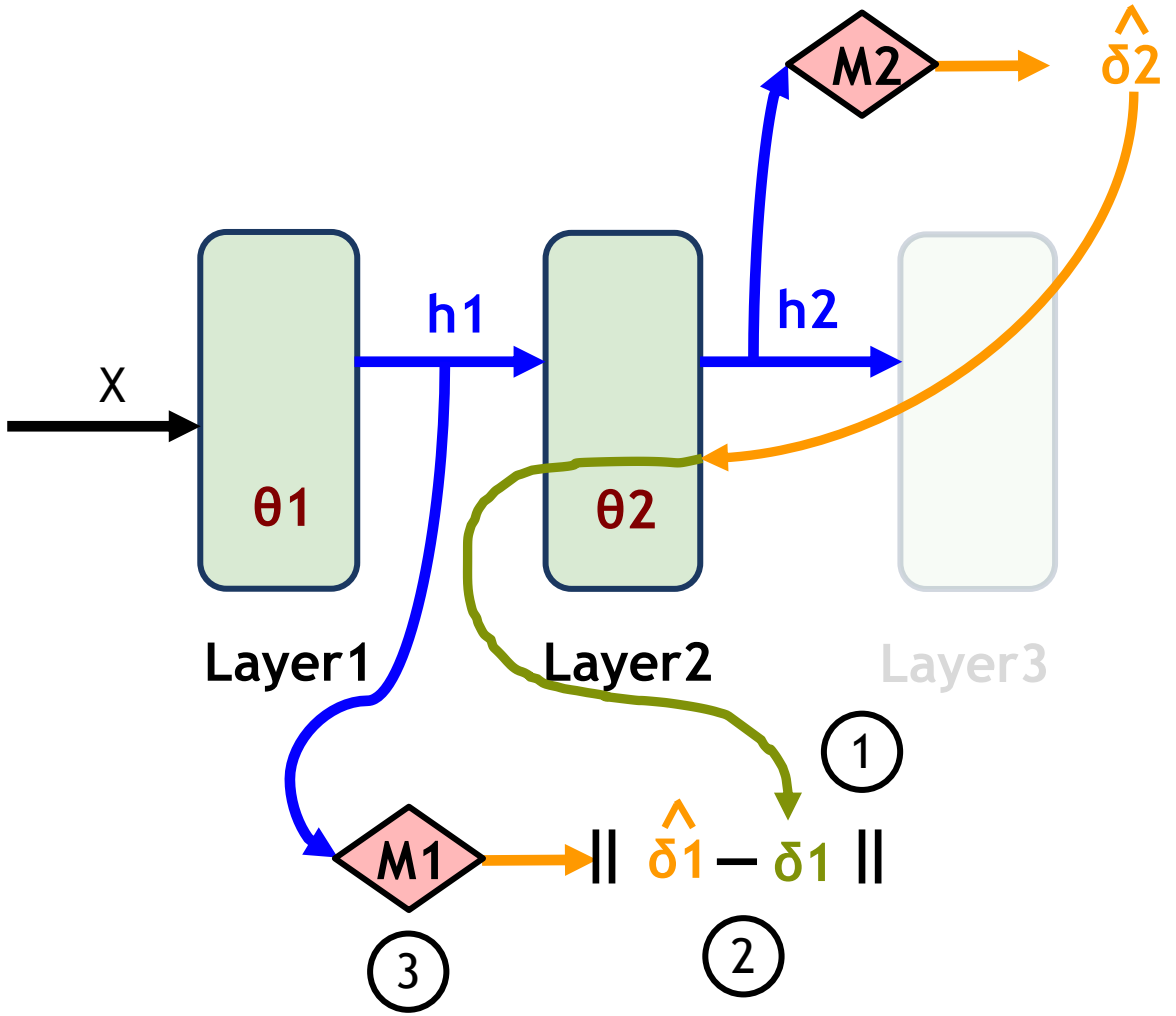
① Find the target -> δ1

② Calculate the loss between $\hat{\delta 1}$ and δ1

③ Update parameters of M1

Repeat Step 1 to 3 until the loss is minimized

Gradient prediction



① Find the target -> δ1

② Calculate the loss between δ1 and δ1

③ Update parameters of M1

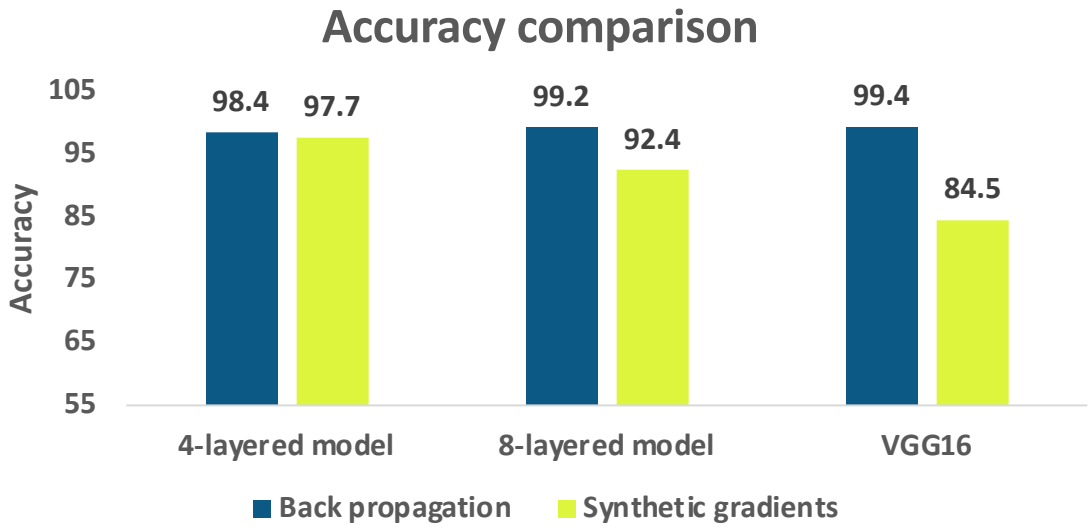
Repeat Step 1 to 3 until the loss is minimized

Evaluation

- Accuracy
  - How accurate is the synthetic gradients approach?
- Convergence speed
  - How fast can the model reach satisfactory accuracy?
- Training speed
  - How much speedup in training when deployed?

Accuracy evaluation

- Using synthetic gradients to achieve model parallelism
  - 4-layered model
    - One convolutional layer and three fully-connected
  - 8-layered model
    - Five convolutional layers and three fully-connected
  - VGG16
  - MNIST Dataset
  - 500K Iteration of training

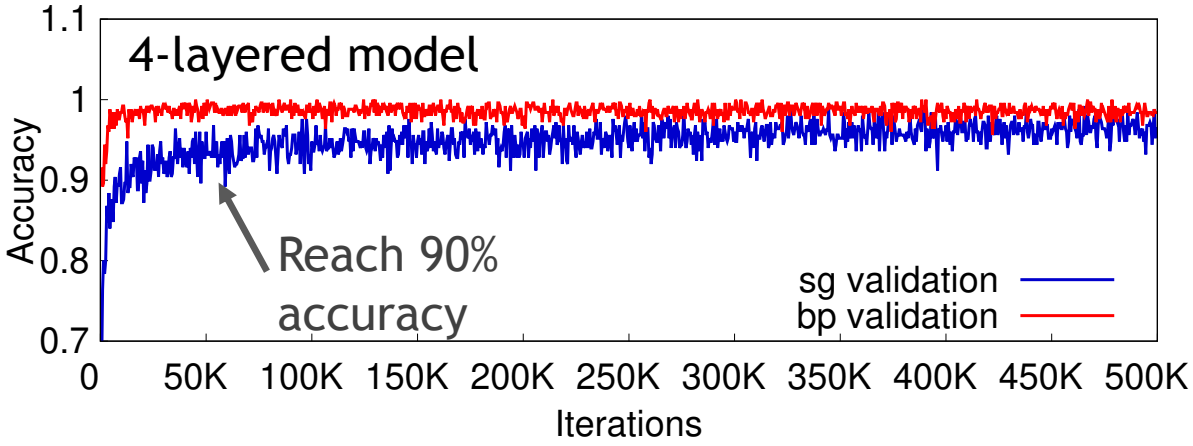


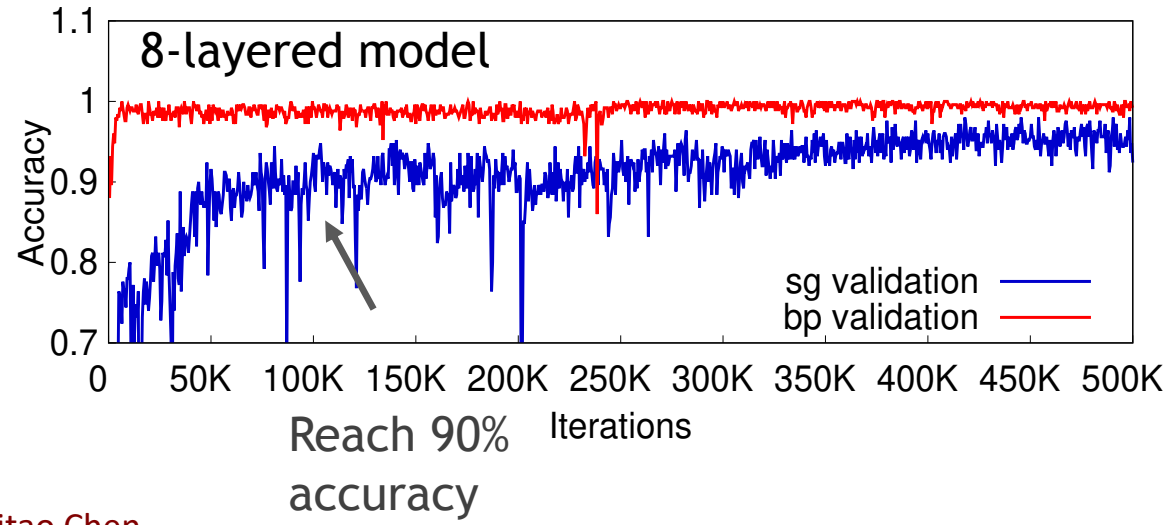
- Effective when the model is shallow
  - 4-layered model: 1%
  - 8-layered model: 5%
- Less effective when the model is complex
  - VGG16: 15%

Convergence speed evaluation

- Synthetic gradients convergence speed
  - 4-layered model
    - One convolution layer and three fully-connected
  - 8-layered model
    - Five convolution layer and three fully-connected
  - MNIST Dataset
  - 500K Iteration of training

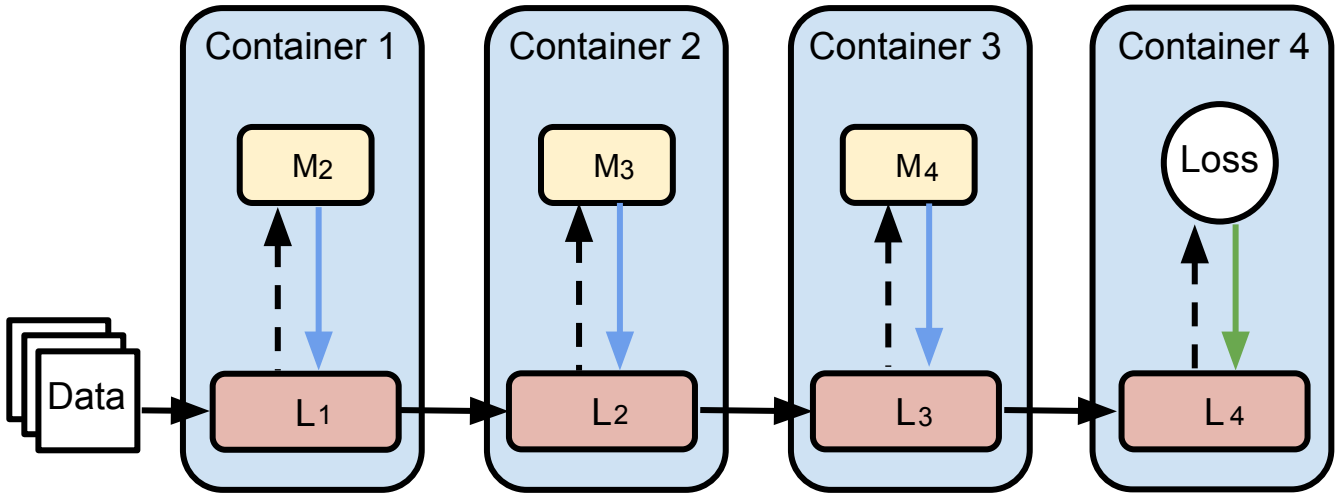
Model complexity slows down the convergence





Training speed evaluation

- Deploy our 4-layered model layer-wise on four Docker container instances
- Use gRPC for communication between instances



- Achieve 11% training speedup
- Each iteration is faster if each layer is trained in parallel

Conclusions

- Need to rethink DNN platforms to handle future learning needs
  - Enable model parallelism without modifying the learning models
- Model parallelism is critical for utilizing cloud and edge resources
  - Synthetic gradients is a feasible solution
- The use of synthetic gradients improves training speed
  - At the cost of some loss in accuracy and convergence speed

Open questions and future work

- How to improve the accuracy when using SG?
  - Explore different gradient prediction models
- How to appropriately partition a model to different devices?
  - Investigate performance difference between model configurations (type, width) and partition based on the available device resources
- How to design a parameter server suited for the SG approach?
  - Investigate compression techniques to minimize communication and provide fast access for SG models
- …

Acknowledgement

- National Science Foundation
  - GEARS project: CNS-1629888
  - CNS-1619653, CNS-1562837, CMMI-1610282, IIS-1633381

- VISA Lab @ ASU
  - Saman, Qirui, Runyu and other lab mates
- Thank you!
  - Questions and suggestions?
  - Come to our poster!



http://gears.asu.edu/

