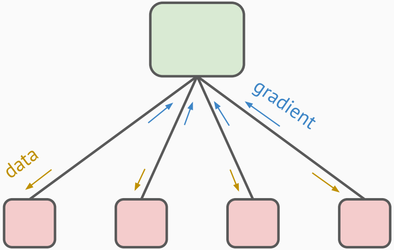
Understanding the effects of data parallelism and sparsity on neural network training

Namhoon Lee[1] Thalaiyasingam Ajanthan[2] Philip H. S. Torr[1] Martin Jaggi[3]

ICLR 2021

[1]University of Oxford [2]Australian National University [3]EPFL

Data parallelism

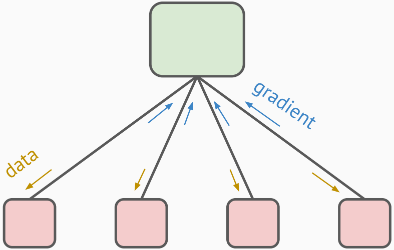


A parallel computing system

Processing training data in parallel

Accelerate training and model-agnostic

Data parallelism



A parallel computing system

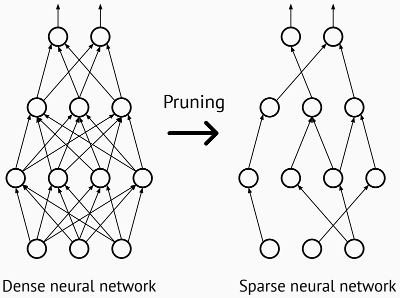
Processing training data in parallel

Accelerate training and model-agnostic

Degree of parallelism $\equiv$ Batch size (single node)

Active research for the effect of batch size (Dean et al. 2012; Goyal et al. 2017; Hoffer et al. 2017; Shallue et al. 2019; Lin et al. 2020)

Sparsity

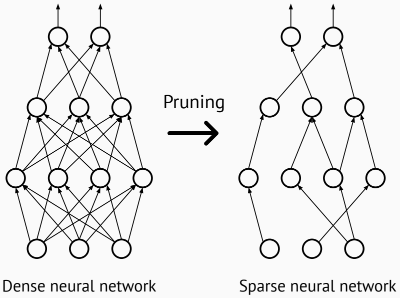


Introducing sparsity by pruning

Sparse neural networks

Save computations and memory

Sparsity



Introducing sparsity by pruning

Sparse neural networks

Save computations and memory

Pruning at initialization prior to training (Lee et al. 2019; Wang et al. 2020)

Subsequent training remains unknown.

Proposal

Data parallelism & Sparsity

- Efficient deep learning
- Complimentary benefits

Proposal

Data parallelism & Sparsity

- Efficient deep learning
- Complimentary benefits

What we do:

1. Measure their effects on training time
2. Develop theoretical analysis to explain the effects

Setup

For a given workload

(a) Network **(b)** Data set **(c)** Algorithm

Train for batch sizes and sparsity levels

Measure steps-to-result ($K^{\star}$)

Setup

For a given workload

(a) Network **(b)** Data set **(c)** Algorithm

Train for batch sizes and sparsity levels

Measure steps-to-result ($K^{\star}$)

Metaparameter search

- Parameters set before training (*e.g.* learning rate)
- To avoid any assumption on optimal metaparameters
- Search space: preliminary results
- Budget: 100 training trials

Setup

For a given workload

(a) Network **(b)** Data set **(c)** Algorithm

Train for batch sizes and sparsity levels

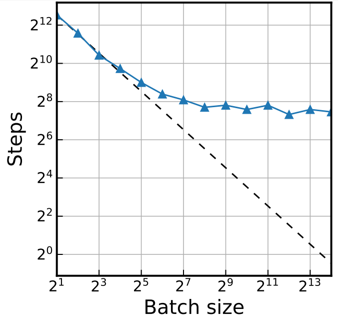
Measure steps-to-result ($K^\star$)

Metaparameter search

- Parameters set before training (*e.g.* learning rate)
- To avoid any assumption on optimal metaparameters
- Search space: preliminary results
- Budget: 100 training trials

Steps-to-result ($K^\star$) vs. Batch size (B)

Measuring the effects

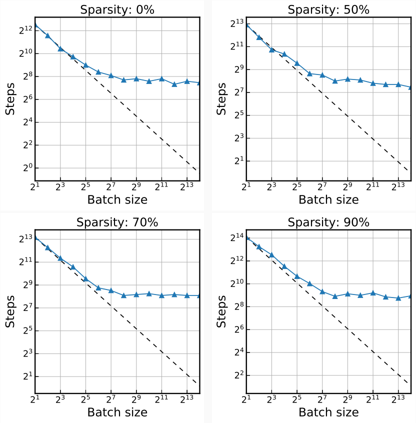


General scaling trend
($K^\star$ vs. B)

General scaling trend across various workloads

- Linear scaling
- Diminishing returns
- Maximal data parallelism

Measuring the effects



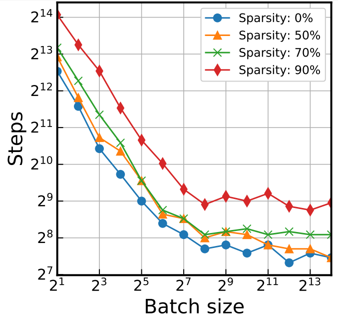
Various sparsity levels

General scaling trend across various workloads

- Linear scaling
- Diminishing returns
- Maximal data parallelism

Sparsity levels (0 – 90%)

Measuring the effects



All sparsity levels

General scaling trend across various workloads

- Linear scaling
- Diminishing returns
- Maximal data parallelism

Sparsity levels (0 – 90%)

Difficulty of training under sparsity

Understanding the effects

Based on convergence properties of stochastic gradient methods:

The relationship between steps-to-result ($K^\star$) and batch size (B)

$$K^\star \approx \frac{c_1}{B} + c_2\,, \qquad \text{where } \; c_1 = \frac{\Delta L \beta}{\mu^2 \varepsilon^2} \; \text{ and } \; c_2 = \frac{\Delta}{\bar{\eta}^\star \mu \varepsilon}\,.$$

Understanding the effects

Based on convergence properties of stochastic gradient methods:

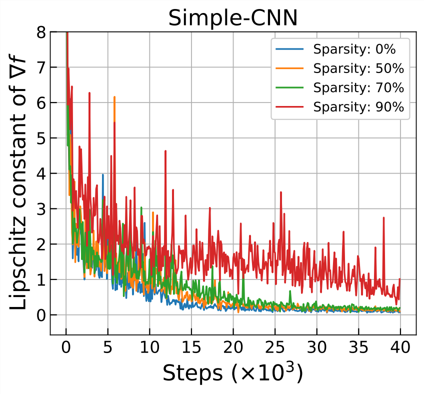
The relationship between steps-to-result ($K^\star$) and batch size (B)

$$K^\star \approx \frac{c_1}{B} + c_2 \,, \qquad \text{where } c_1 = \frac{\Delta L \beta}{\mu^2 \varepsilon^2} \text{ and } c_2 = \frac{\Delta}{\bar{\eta}^\star \mu \varepsilon} \,.$$

This result precisely illustrates the observed scaling trends.

1. Linear scaling, diminishing returns, maximal data parallelism
2. Lipschitz smoothness (L) is what can shift the curve vertically

Lipschitz smoothness under sparsity



Local L throughout training

Local Lipschitz smoothness ($\boldsymbol{L}$)

The higher sparsity, the higher $\boldsymbol{L}$

Gradient changes relatively too quickly

The difficulty of training sparse networks

Summary

Main points:

1. General scaling trend for the effects of data parallelism and sparsity
2. Theoretical analysis to verify the effects
3. Lipschitz smoothness to explain the difficulty of training sparse networks

Code: https://github.com/namhoonlee/effect-dps-public

Contact: namhoon@robots.ox.ac.uk

References i

Dean, Jeffrey et al. (2012). “Large scale distributed deep networks”. In: *NeurIPS*.

Goyal, Priya et al. (2017). “Accurate, large minibatch sgd: Training imagenet in 1 hour”. In: *arXiv preprint arXiv:1706.02677*.

Hoffer, Elad, Itay Hubara, and Daniel Soudry (2017). “Train longer, generalize better: closing the generalization gap in large batch training of neural networks”. In: *NeurIPS*.

Lee, Namhoon, Thalaiyasingam Ajanthan, and Philip HS Torr (2019). “SNIP: Single-shot network pruning based on connection sensitivity”. In: *ICLR*.

Lin, Tao et al. (2020). “Don’t Use Large Mini-Batches, Use Local SGD”. In: *ICLR*.

Shallue, Christopher J et al. (2019). “Measuring the effects of data parallelism on neural network training”. In: *JMLR*.

Wang, Chaoqi, Guodong Zhang, and Roger Grosse (2020). “Picking Winning Tickets Before Training by Preserving Gradient Flow”. In: *ICLR*.