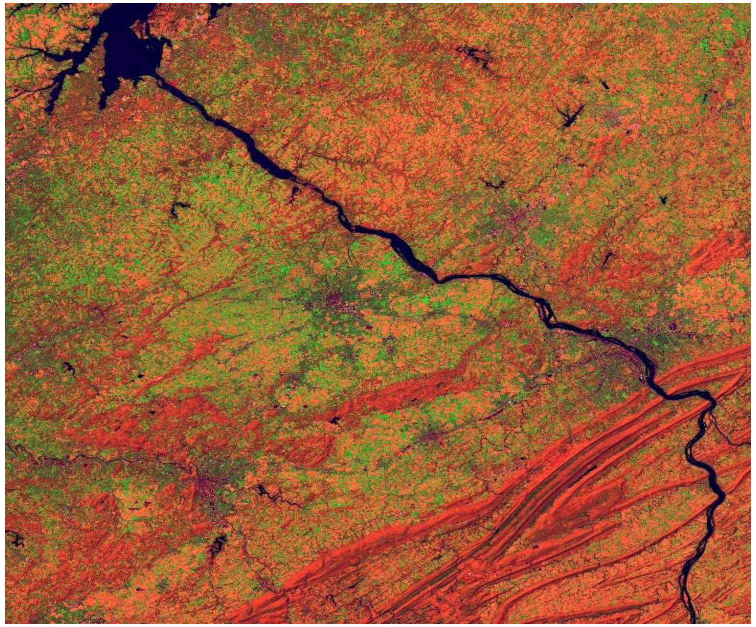


ENHANCING LARGE BATCH SIZE TRAINING OF DEEP MODELS FOR REMOTE SENSING APPLICATIONS

Rocco Sedona[1,2], *Gabriele Cavallaro*[1], *Morris Riedel*[1,2], *Matthias Book*[1]

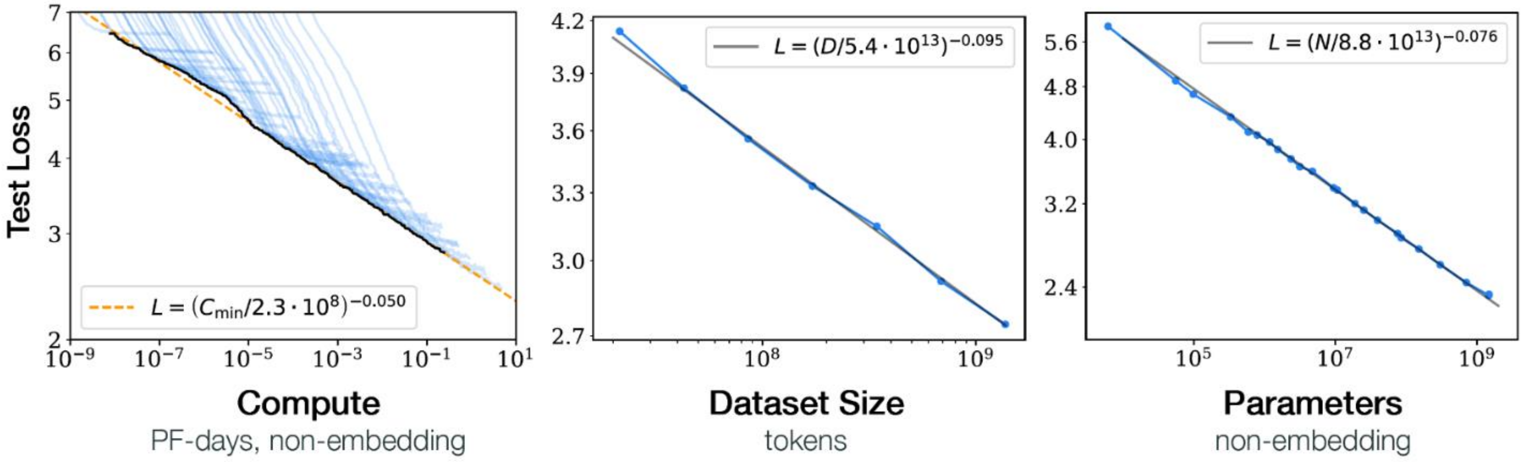
[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
[2] School of Engineering and Natural Sciences, University of Iceland, Iceland

JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

UNIVERSITY OF ICELAND
SCHOOL OF ENGINEERING AND NATURAL SCIENCES
FACULTY OF INDUSTRIAL ENGINEERING, MECHANICAL ENGINEERING AND COMPUTER SCIENCE

Distributed Deep Learning

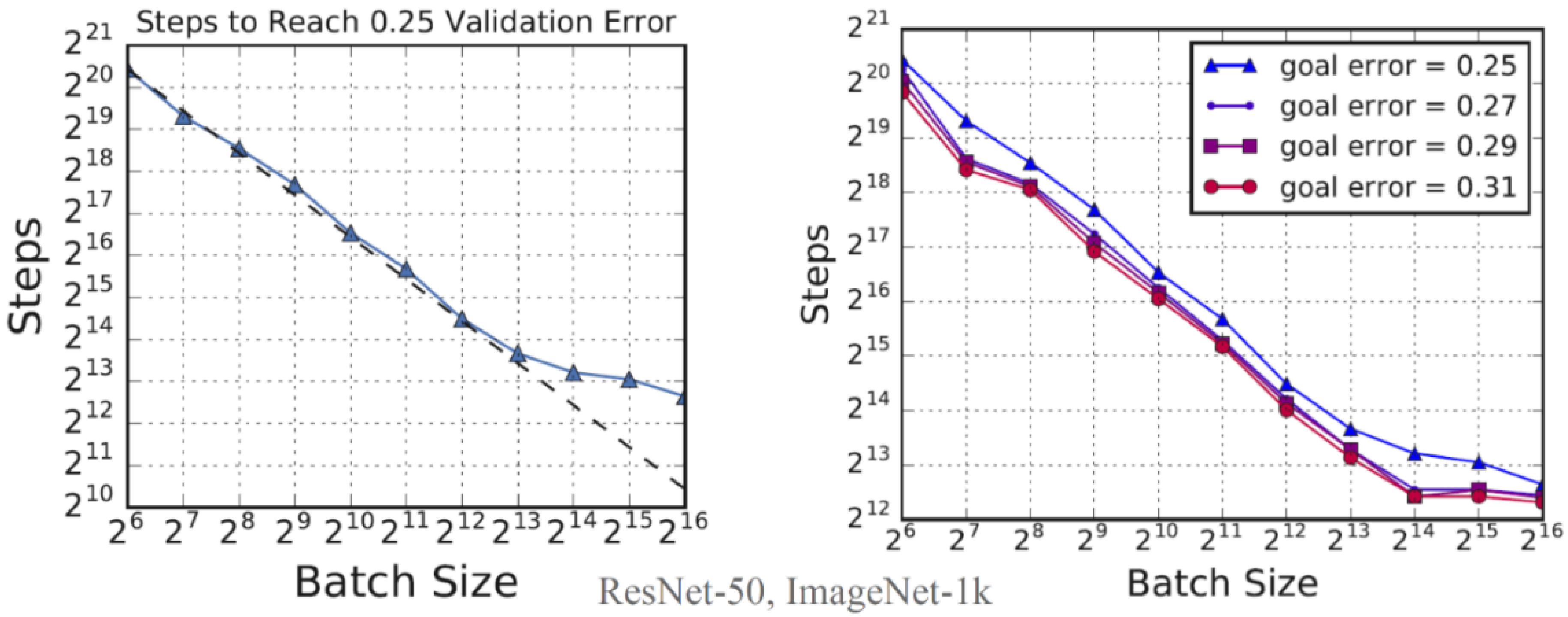
- Trend is to train larger models
- Larger models require larger datasets
- How? Data parallelism on HPC resources comes at aid
- In data parallelism a model is replicated on N GPUs
- Different chunks of data on each GPU
- Resulting global batch size is $B_{global} = B_{local} \times N$



***[1]** Large models and datasets*

Challenges of Distributed Learning

- At a given batch size SGD stops to scale
- The number of steps to a given accuracy does not decrease anymore

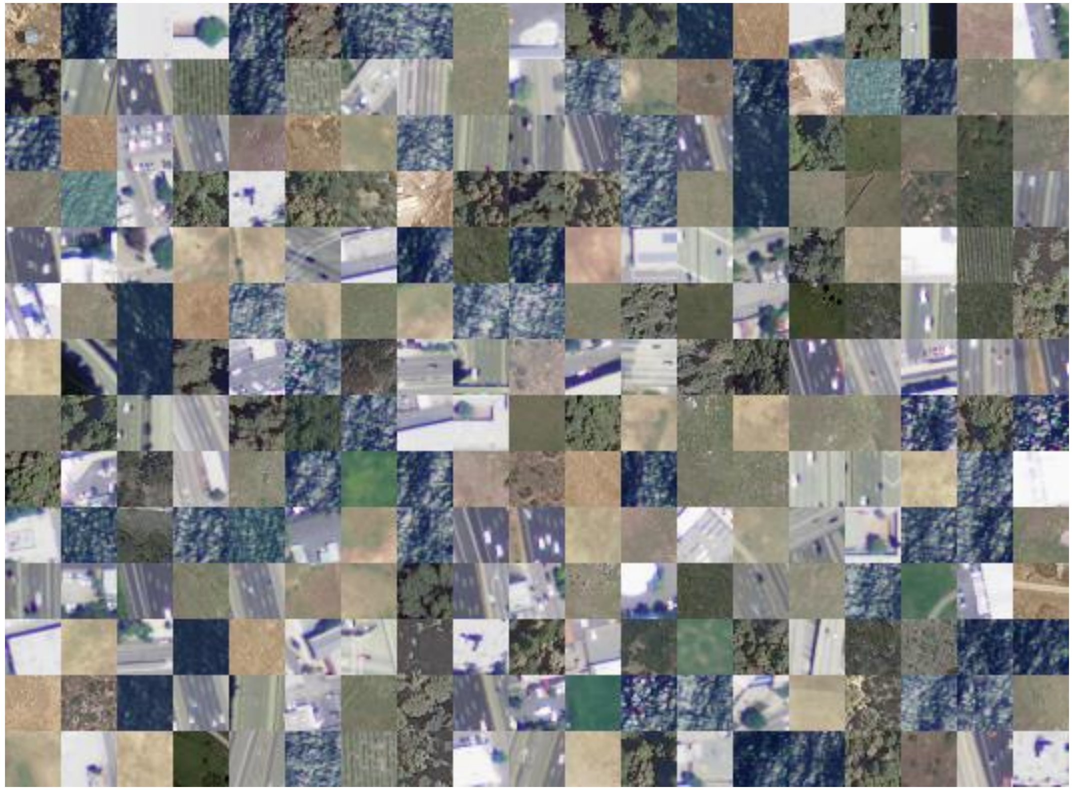


[2] Steps to accuracy

Dataset

- Extracted from disjoint NAIP tiles
- 28x28 pixels
- 4 channels (RGB + Near IR)
- 1 m spatial resolution
- 80% training, 20% test

- SAT-4
  - consists of a total of 500,000 image patches covering four broad land cover classes
- SAT-6
  - consists of a total of 405,000 image patches each of size 28x28
  - covering 6 landcover classes
  - 4 channels (RGB + Near IR)



***[3]** SAT4 and SAT6*

Training Strategy

- Model: ResNet50
  - Skip connections to reduce vanishing gradient
- LAMB optimizer [4]

$$x_{t+1}^{i} = x_{t}^{i} - \eta_t \frac{\Phi(\|x_t^i\|)}{\|g_t^i\|} g_t^i,$$

- Warm-up phase scaled w.r.t. the batch size
- Root square policy for initial learning rate
- Polynomial learning rate scheduler

Experimental Setup

- DEEP-EST supercomputer at JSC
- Extreme Scale Booster Partition (ESB)
- Up to 32 GPUs (Nvidia V100)
- Horovod on top of TF2 (Keras API)
  - MPI vs NCCL
- 100 epochs
- Simple data augmentation techniques
- 3 runs for each experiment

[5] DEEP-EST

Results

- Test accuracy satisfactory up to batch size of 32K
- Training time is reduced (although less than linear)
- Test loss increases with the increase of the batch size
- Significant divergence at 65K

Batch size	N. GPUs	Accuracy	Loss	Time [s]
8K	4	0.99	0.02	34
16K	8	0.98	0.07	18
32K	16	0.96	0.11	9
65K	32	diverges		5

Table 2. Accuracy and test loss, training time per epoch epoch with LAMB optimizer, dataset SAT4.

Batch size	N. GPUs	Accuracy	Loss	Time [s]
8K	4	0.99	0.05	41
16K	8	0.98	0.11	22
32K	16	0.94	0.17	11
65K	32	diverges		6

Table 3. Accuracy and test loss, training time per epoch epoch with LAMB optimizer, dataset SAT6.

Comments and Future Developments

- Training scaled on large number of GPUs and training time reduced
- Comparison MPI vs NCCL
- Fair comparison with other optimizers [6]
  - Utilization of evolutionary optimization
- Thorough analysis of steps to accuracy
- Extend to more complex datasets

References

[1] Kaplan et al., “Scaling Laws for Neural Language Models”, 2020, https://arxiv.org/abs/2001.08361

[2] Shallue et al., Measuring the Effects of Data Parallelism on Neural Network Training”, JMLR, 2020

[3] SAT4 and SAT6, https://csc.lsu.edu/~saikat/deepsat/

[4] You et al., “Large Batch Optimization for Deep Learning: Training BERT in 76 minutes,” 2020

[5] DEEP Projects, https://www.deep-projects.eu/

[6] Nado et al., A Large Batch Optimizer Reality Check: Traditional, Generic Optimizers Suffice Across Batch Sizes , https://arxiv.org/pdf/2102.06356.pdf