Efficient Distributed Deep Learning using Circulant Matrices

Alexandre Araujo[1] Benjamin Negrevergne, Jamal Atif

Université Paris-Dauphine, PSL Research University, CNRS, LAMSADE, 75016 Paris, France

Wavestone, Paris, France

[1]alexandre.araujo@wavestone.com

Deep Learning

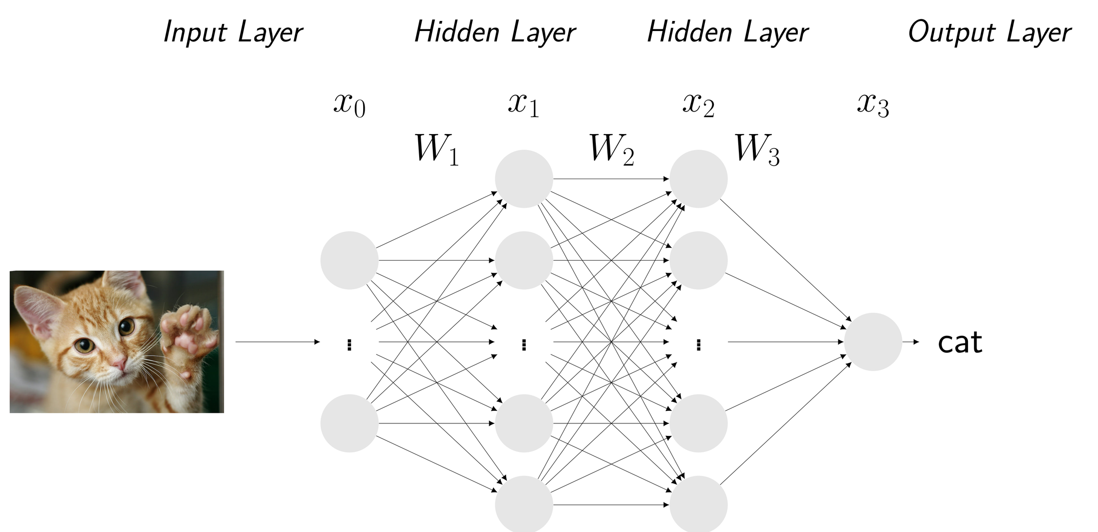

Figure: Fully Connected Neural Network with two hidden layers.

W_i matrices can be very large:

- **millions of parameters**
- **several gigabytes of memory**

Distributed Deep Learning

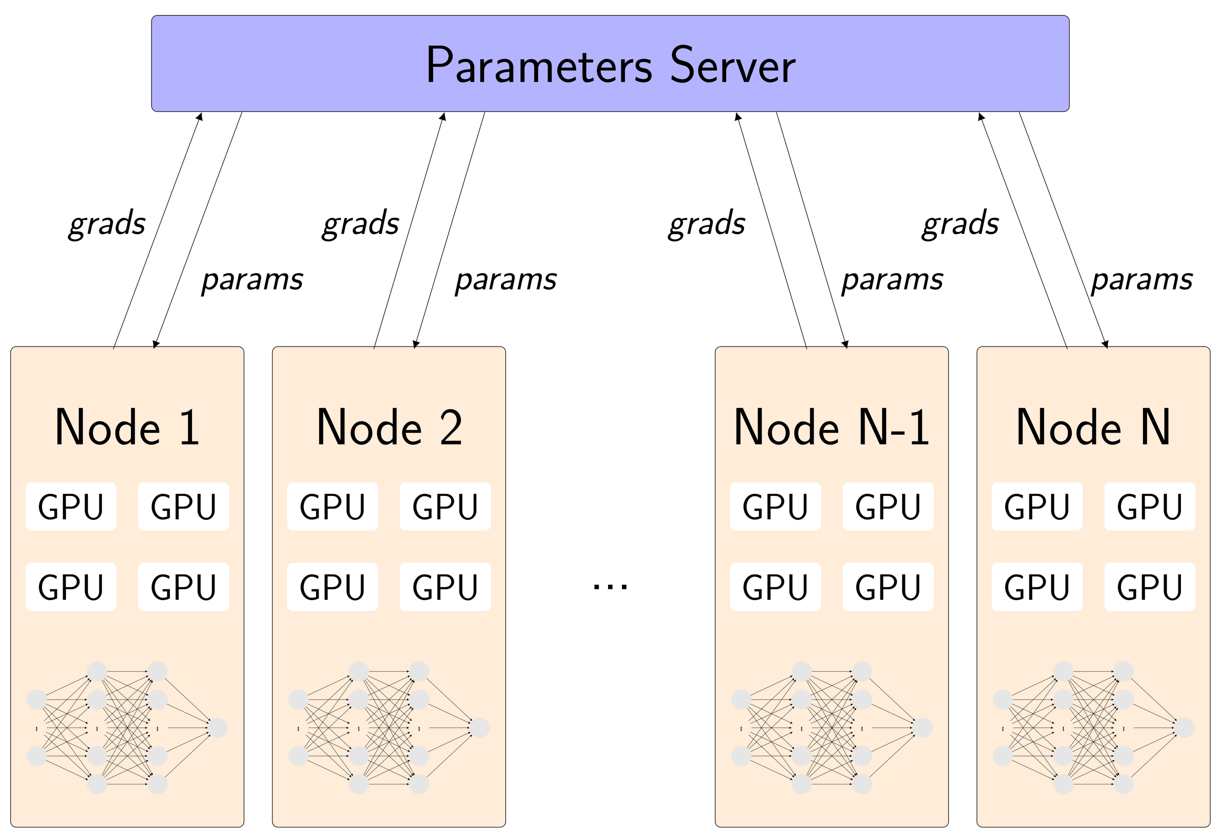

Figure: Asynchronous Synchronization of different nodes with the parameters server. The parameters and the gradients needs to traverse the network.

Distributed computing allows fast training of NN but given the amount of parameters in recent very deep network architectures, **a large bandwidth** is needed to transfer the gradients and parameters updates.

Reducing network communication

Terngrad [1]: Gradients are quantized into ternary precisions before being sent to the parameters server: $\{-1, 0, 1\}$.

Sufficient Broadcasting [2]: the parameters update ∇W is a *low-rank* matrix and can be written $\nabla W = uv^T$. The matrix is decomposed before being sent to the parameters server.

Those techniques can reduce memory footprint of parameters and/or gradients which reduces communication time between *workers* and *Parameters server*. This comes with some limitations:

- additional operations on top of the training,
- techniques independent of the training.

Main Idea: Using Circulant Matrices to reduce network bandwidth

A circulant matrix $C \in \mathbb{R}^{n \times n}$ can be defined by:

$$C = circ(c) = \begin{pmatrix} c_0 & c_1 & c_2 & \dots & c_{n-1} \\ c_{n-1} & c_0 & c_1 & & c_{n-2} \\ c_{n-2} & c_{n-1} & c_0 & & c_{n-3} \\ \vdots & & & \ddots & \vdots \\ c_1 & c_2 & c_3 & & c_0 \end{pmatrix}$$

with $c \in \mathbb{R}^n$ and $c = [c_0, c_1, c_2, \dots, c_{n-1}]$.

Circular convolution Theorem

$$C \cdot x = \mathfrak{F}^{-1}(\mathfrak{F}(c) \times \mathfrak{F}(x))$$

With $x \in \mathbb{R}^n$ and $\mathfrak{F}$ denotes to the *Fourier Transform*.

Main Advantages

- **Reduce the memory** required of the parameters (only the redundant vector has to be stored in memory).
- Circulant matrices can be **trained end-to-end** to improve performances.
- **Reduce the complexity** with the use of *FFT* algorithm.
- **Reduce the amount of communication** between nodes with distributed computing.

Preliminary Results

Experiments realized with a **5 layers architecture**:

- 2 convolution layers
- 3 fully connected layers with either dense or circulant matrix

Type	#Params	Compress. (%)	Precision (%)
dense	280 464	-	**99.14**
circulant	14 224	94.92	98.55

Table: Result on MNIST Dataset

Type	#Params	Compress. (%)	Precision (%)
dense	1 068 298	-	**84.2**
circulant	112 896	89.43	80.0

Table: Result on CIFAR10 Dataset

Future work

- Improve and understand the training and convergence NN with Circulant Matrices
- Use Circulant Matrices on large scale distributed computing
- Investigate other structured matrices.

[1] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 1509–1519, Curran Associates, Inc., 2017.

[2] P. Xie, J. K. Kim, Y. Zhou, Q. Ho, A. Kumar, Y. Yu, and E. Xing, "Lighter-communication distributed machine learning via sufficient factor broadcasting," in *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, (Arlington, Virginia, United States), pp. 795–804, AUAI Press, 2016.

[3] Y. Cheng, F. X. Yu, R. S. Feris, S. Kumar, A. Choudhary, and S. F. Chang, "An exploration of parameter redundancy in deep networks with circulant projections," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2857–2865, Dec 2015.