A System for Studying Deep Network Training

Raul Castro Fernandez
The University of Chicago
raulcf@uchicago.edu

The unprecedented performance of deep neural networks (DL) remains hard to explain: why do neural networks perform so well in so many tasks? Understanding the learning dynamics of DL would open a myriad of opportunities to both improve optimization algorithms, as well as to gain theoretical understanding of why neural networks work so well.

Having a theoretical understanding of why DL works is a crucial milestone. Not only intellectually, but also practically. For example, if we knew how they work, we would have a better understanding of their limits, and this would open an avenue to decide what problems are suitable for DL and what problems are not, hence saving lots of resources and human labor.

Because gaining insight on how DL works is so important, many approaches and theories have been proposed, from modeling networks as a spin-glass model [1] of statistical mechanics as early as 1985, to more modern attempts to model them using mutual information and entropy [2], or the theory of random matrices [4]. In all these approaches, researchers propose a hypothesis along with a theory to explain the hypothesis. Then, they study small networks due to the complexity of larger, real ones, in order to understand whether the hypothesis holds or not. All these approaches follow a common practice in the physical sciences: to propose a model that approximates a large complex system so it can be studied in detail.

In this project, we propose instead to derive the hypotheses in a data-driven way [3]. In particular, we propose to gather all intermediate data produced while training a neural network to then study it offline. With all data available, it becomes possible to test the hypothesis other researchers are proposing more efficiently. By analyzing the dynamics data, we can gain valuable insights that remain hidden, and would allow us to propose hypotheses that explain how certain aspects of training work. In other words, by possessing all data, we can both test existing hypotheses as well as generate new ones from the data, speeding up discovery.

Many challenges spanning theory, data structures, approximation algorithms and systems are in the way. The main challenge concerns the huge amount of computing resources—storage, memory, and compute—that collecting and processing this data entails. For example, a *small* network with only 500K parameters and a training dataset with 100K samples requires 223GB of space to record how every parameter changes with each input sample during a single *training session* of 50 iterations. To understand learning dynamics, we may be interested in executing several different training sessions and understand the differences (and similarities) between them—for example, to understand how different orderings of the input data, or how different weight initializations affect convergence. When we need to store the data for several training sessions, storage needs quickly increase.

Despite the large storage needs of the approach, perhaps the major challenge is in compute and memory. Next, we offer a small sample of queries that we wish to run:

• Show what group of samples caused each weight to change the most during the epoch number X?
• Group the weights that have changed the most (largest gradients) during the entire training process.
• For two different executions, show how much learning progress a network has made by the time they have executed the N step, given two different orderings of input samples.
• What were the sequence of last-K input samples when more than 10% gradients were about to explode?
• What was the sequence of K-input samples that lead to this weight to become zero?

These are only a very small sample of queries that are needed to test even the simplest hypothesis of learning dynamics. These few queries, however, suffice to motivate the design and implementation of efficient index structures to support these queries across the modern memory hierarchies. At the same time, they motivate the design of new algorithms for the efficient processing of such queries. To make this a reality consists of more than just designing the right index structures. We need efficient methods to obtain the data while the network is training at the same time we avoid overheads and bottlenecks. In order to allow others to pose different questions, we need a form of domain-specific language.

All in all, the challenges of studying deep networks justify a new class of systems: in addition to systems for training deep networks, and systems to deploy deep networks, in this project we argue for systems for studying them.

With such a grand challenge in front of us, extracting the maximum value out of the hardware available to us today—storage, memory, CPU—is crucial. The consequences of gaining understanding on how DL works justify the investment in time and research effort.

Last, because neural networks percolate a myriad of applications today, and they are set on spreading even more, their heterogeneity means that their study will become a long process. How do we represent batch processing? And recurrent neural networks? dropout? What about network that use more sophisticated mechanisms such as memory? In principle, all these variations are introducing even more state, therefore increasing the challenge. More people and teams will be necessary to take on this challenge, and this may motivate the design of specific hardware architectures deemed specifically for studying deep neural networks.

REFERENCES

[1] D. Amit, H. Gutfreund, et al. Spin-glass models of neural networks. In *Physical Review A*, 1985.
[2] M. Gabrie, A. Manoel, et al. Entropy and mutual information in models of deep neural networks. In *NeurIPS*, 2018.
[3] T. Hey and S. Tansley. The fourth paradigm: Data-intensive scientific discovery. In *Microsoft Research*, 2009.
[4] J. Pennington and P. Worah. Nonlinear random matrix theory for deep learning. In *NIPS*, 2017.