DAWNBench: An End-to-End Deep Learning Benchmark and Competition

Cody Coleman, Deepak Narayanan, Daniel Kang, Tian Zhao, Jian Zhang, Luigi Nardi, Peter Bailis, Kunle Olukotun, Chris Ré, Matei Zaharia

Stanford DAWN Project
http://dawn.cs.stanford.edu/benchmark

Abstract

Despite considerable research on systems, algorithms and hardware to speed up deep learning workloads, there is no standard means of evaluating end-to-end deep learning performance. Existing benchmarks measure proxy metrics, such as time to process one minibatch of data, that do not indicate whether the system as a whole will produce a high-quality result. In this work, we introduce DAWNBench, a benchmark and competition focused on end-to-end training time to achieve a state-of-the-art accuracy level, as well as inference with that accuracy. We have seeded the benchmark with entries for image classification on CIFAR10 and ImageNet, and question answering on SQuAD, showing differences across models, software and hardware. We believe DAWNBench will provide a useful, reproducible means of evaluating the many tradeoffs in deep learning systems.

1 Introduction

Deep learning methods are effective but computationally expensive, leading to a great deal of work to optimize their computational performance. Researchers have proposed new software systems [1, 7, 8, 11, 23, 39], training algorithms [12, 15, 21, 22, 26, 35–38, 40, 42], communication methods [8, 10, 11, 19, 32, 41] and hardware [6, 16–18, 24, 30] to decrease this cost. Despite significant advances, it is hard to measure or compare the utility of these results due to a lack of standard evaluation criteria. Most existing benchmarks for deep learning performance [2–4, 7, 9, 14, 34] only measure proxy metrics such as the time to process one minibatch of data. In reality, deep learning performance is far more complex. Approaches such as using larger batch sizes [15, 24], reduced precision [8, 10, 18, 20] and asynchronous updates [8, 11, 32, 41] can stop an algorithm from converging to a good result, or increase the time to do so. These approaches also interact in nontrivial ways and may require updating the underlying optimization algorithm [15, 26, 29], further affecting performance.

This lack of standard evaluation criteria leaves deep learning practitioners having to navigate these trade-offs. For example, minimal effort back propagation (meProp) delivers a 3.1x speed up over back propagation on MNIST [36]. Using 8-bit precision gives a 3x speed up on MNIST [10]. Does combining meProp with 8-bit precision give a 9.3x speed up? Would that speed translate to a larger model on a dataset like ImageNet, and combine with accurate, large minibatch SGD [15] to train an ImageNet model in 7 minutes? Currently, these questions can only be answered via tedious and time-consuming experimentation. Researchers face a similar challenge: when they have a new idea for an optimization, which previous techniques should they consider combining in evaluating their results?

To provide an objective means of quantifying end-to-end deep learning performance, we introduce DAWNBench, an open benchmark and competition for end-to-end deep learning training and inference. Instead of simply measuring time per iteration (or throughput), DAWNBench measures end-to-end performance in training (e.g., time, cost) and inference (e.g., latency, cost) at a specified state-of-the-art level of accuracy. This provides an objective means of normalizing across differences in computation frameworks, hardware, optimization algorithms, hyperparameter settings, and other factors that affect real-world performance. Our initial release of DAWNBench provides end-to-end learning and inference tasks including image classification on CIFAR10 [27] and ImageNet [33], and question answering on SQuAD [31], and reference implementations for each task. Over time, with community input, we plan to expand the set of benchmark tasks (e.g., segmentation, machine translation, video classification) and metrics (e.g., power, sample complexity).

2 Benchmark Structure

DAWNBench evaluates deep learning systems on different tasks based on several metrics. The benchmark allows innovation in software, algorithms, communication methods, etc. By only specifying the task, DAWNBench also allows

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image classification</td>
<td>Training time</td>
</tr>
<tr>
<td>Question answering</td>
<td>Training cost</td>
</tr>
<tr>
<td></td>
<td>Inference latency</td>
</tr>
<tr>
<td></td>
<td>Inference cost</td>
</tr>
</tbody>
</table>

Table 1: Dimensions evaluated in the first version of DAWNBench. All metrics are for a near-state-of-the-art accuracy.
experimentation of new model architectures and hardware. In the v0 release, we seed entries for two tasks: image classification on CIFAR10 and ImageNet, and question answering on SQuAD, and evaluate on four metrics: training time to a specified validation accuracy, cost of training to a specified validation accuracy, latency of performing inference on a single item (image or question), and cost of inference for a single item (Table 1). We intend to update the benchmark with more tasks and new accuracy targets over time.

3 Example Results

In this section, we offer preliminary results that seek to answer two questions: (1) Is training time to a specified validation accuracy a useful metric to evaluate deep learning systems? (2) What type of insights can DAWNBench surface?

Evaluating Impact of Minibatch Size. To illustrate the value of DAWNBench’s end-to-end performance metric, we use it to study how minibatch size impacts both the convergence rate and hardware performance (FLOPS) of a deep learning workload, making it hard to reason about end-to-end performance from either metric alone. Prior work [5, 13, 15, 25, 28] has shown that picking a minibatch size too small or too large can lead to poor convergence, i.e. minibatch size affects convergence. Additionally, larger minibatch sizes better saturate hardware execution units [5, 13]. In choosing the minibatch size that minimizes total time to a target accuracy, we must balance these two factors. As we show in Figure 1, for a ResNet56 model trained on the CIFAR10 dataset on a Nvidia P100 GPU, a minibatch size of 32 produces the best convergence rate (least number of epochs to highest accuracy), and a minibatch size of 2048 produces the best throughput (number of images processed divided by total time taken). A minibatch size of 256 represents a reasonable trade-off between convergence rate and throughput. A minibatch size of 256 reaches an accuracy of 93.38%, which is only 0.43% less than the maximum accuracy achieved with a minibatch size of 32, in 1.9x less time. Benchmarks that focus exclusively on convergence rate and throughput are unable to surface these practical trade-offs for factors even as simple as minibatch size.

Comparison of DAWNBench Seed Entries. We seeded DAWNBench with single-GPU and CPU results for TensorFlow and PyTorch, using reference implementations of models when possible. We show some of the variability present across DAWNBench’s metrics even from simple factors such as the model, software framework and hardware type in Figure 2. This figure presents training time to 93% validation accuracy, and single-image inference latency for various ResNet architectures for the CIFAR10 dataset, on different hardware platforms (1 K80 GPU on two cloud providers [Google and Amazon], 1 P100 GPU on a private cluster, and a 16vCPU machine on Google Cloud) and frameworks.

As the figure illustrates, TensorFlow is faster than PyTorch on CPUs, but slightly slower on GPUs, both for training and inference. This is partly due to data format: TensorFlow supports both NCHW and NHWC layouts (N: Number of Samples, C: Number of Channels, H: Height, W: Width), which give better performance on GPUs and CPUs respectively, while PyTorch only supports NCHW. K80 performance is similar on both cloud providers, but with spot pricing for GPU instances, Amazon is cheaper. Training and inference time are proportional to the depth of the model, as expected.

4 Conclusion

DAWNBench proposes a simple, living benchmark for the performance metrics practitioners care about most: end-to-end time to train a model with state-of-the-art accuracy, and inference time with that accuracy. We hope that this collection of tasks, seed entries and our ongoing competition will provide a simple way to test and validate a wide variety of new ideas, spanning systems, algorithms, and hardware, to optimize deep learning. We intend to keep DAWNBench up to date with new tasks and goals to help the community track progress in deep learning systems.
Acknowledgments

We thank the many members of the Stanford InfoLab for their valuable feedback on this work. This research was supported in part by affiliate members and other supporters of the Stanford DAWN project – Intel, Microsoft, Teradata, and VMware – as well as industrial gifts and support from Toyota Research Institute, Juniper Networks, Keysight Technologies, Hitachi, Facebook, Northrop Grumman, NetApp, and the NSF under grants DGE-1656518, DGE-114747, and CNS-1651570.

References


