Does TensorFlow 2 capably rival PyTorch?

Technology Review | Devin Burke, Author

Introduction

In 2018, Facebook's PyTorch was a small player in the machine-learning field compared to Google's first-in-class TensorFlow (He, 2019). But it had one advantage: better performance and ease of use. Just one year later, TensorFlow was hemorrhaging market share. When Google announced TensorFlow 2 in 2019, use of PyTorch had already surpassed it in academic research, a crucial domain for AI technology adoption. Still, implied in the announcement, TensorFlow 2 was intended as a corrective integrating those same features that had led people away from its 1.x version (TensorFlow Team, 2019).

Here, one can draw parallels with the history of the JavaScript framework Angular: Google's AngularJS was ascending the web-application space until Facebook's ReactJS began displacing it with a more native-JavaScript integration. Google responded by rewriting AngularJS as Angular 2 using a similar native-JavaScript approach, but also incorporating features missing its Facebook-backed competitor (Gavigan, 2018).

But is this comparison appropriate? Does TensorFlow 2 adequately bridge the gap between the itself and PyTorch?

Background

Origin of TensorFlow

In 2011, the Google Brain team developed a machine-learning infrastructure called DistBelief to be used internally for deep-learning neural networks (Dean, Monga, et al., 2015). Google Brain itself is a research team within the company's overall artificial intelligence R&D division dedicated to the exploration and exploitation of machine learning and natural language processing. The project was a success, as DistBelief was quickly incorporated into many internal systems including the Google Image search function. By 2015, Google deprecated DistBelief as a first-generation technology and succeeded it with the TensorFlow library under the direction of leading computer scientists.

TensorFlow became the machine-learning backbone to numerous Google products. In the public announcement of TensorFlow, Google CEO Sundar Pichai (2015) revealed that the company internally uses it "for everything from speech recognition in the Google app, to Smart Reply in Inbox, to search in Google Photos." More, Google was able to build and train neural networks five times faster with TensorFlow than with DistBelief.

In the same announcement, TensorFlow was released open-source for unlimited commercial applications. This made TensorFlow widely considered first-to-arrive in its generation of open-source, machine-learning frameworks (Oremus, 2015). But as with any firstcomer, it had limitations that a next-to-market framework would have the hindsight to avoid.

Origin of PyTorch

PyTorch has a similar history. It originated in competitor-in-spirit Facebook's AI Research (FAIR) division as the now-deprecated Torch framework, before being re-created as PyTorch in 2016 (Yegulalp, 2017). And like TensorFlow, PyTorch is machine-learning library released under open source licensing, used for natural language processing, graphics processing, and other related artificial intelligence functions.

However, PyTorch originated several features that TensorFlow lacked. Most notably, PyTorch had dynamic graph data consumption—meaning graph structures did not have to be predefined, and data did not have to confirm to a set structure—as well as inherent paralellelism and multi-GPU processing (Yegulalp, 2017).

To add insult to injury, PyTorch had a less verbose, very 'Pythonic' API that many developers simply found more intuitive and preferred for reasons of taste (He, 2019).

Since released, PyTorch was adopted by such heavy hitters as SalesForce, Twitter, Carnegie Mellon, and, of course, Facebook itself.

TensorFlow vs PyTorch

Then: TensorFlow 1.x vs PyTorch

The core function of both technologies is the processing of models as directed graphs. But TensorFlow's 2015 release required static definition of these graphs, to which all data must conform. Further, it had a verbose API and exposed little of its inner components, which was counter to Pythonic conventions, and lacked parallelism natively (Dubovikov, 2017). With PyTorch's 2016 release, TensorFlow's presence in AI research projects slowly dwindled due to the former's dynamic graphing capabilities and flatter learning curve (He, 2019).

In a comparison of TensorFlow 1.x and PyTorch, Kirill Dubovikov (2017), CTO of Cinimex DataLab, illustrated the problem of static graphs: "if you develop a sentiment analysis model for English sentences you must fix the sentence length to some maximum value and pad all smaller sequences with zeros." This leaves little room for real-world variables, such as unpredictably long sentences, to be handled without recompiling the graph model. He further compared TensorFlow 1.x's API channeling all communication through one session interface to having "your model… behind a brick wall with several tiny holes to communicate over."

This interface funneling and lack of inner-component exposure prevent use of native Python debugging—requiring an external tool called *tfdbg* to evaluate TensorFlow expressions (Dubovikov, 2017). PyTorch, on the other hand, integrates natively with Python, is fully dynamic in its graph data consumption and eager execution of code.

Dubovikov points out another key weakness of first-generation TensorFlow: "One of the biggest features that distinguish PyTorch from TensorFlow is declarative data parallelism" because PyTorch "can leverage multiple GPUs with almost no effort" but in TensorFlow "defining parallelism is way more manual and requires careful thought." This means that out-of-the-box TensorFlow code is much slower than that of PyTorch because it does not have as many simultaneous processing resources.

These differences did not go unnoticed by the technology community. In a 2019 summarization of the state of ML frameworks, author Horace He (2019) analyzed references to the two technologies in research-community conferences, and the conclusion was clear: use of PyTorch had surpassed TensorFlow and its momentum was on the incline.

Now: TensorFlow 2.x vs PyTorch

Just as PyTorch was developed with hindsight for TensorFlow's limitations, so too did Google recognize PyTorch's strengths and aim to re-create them. In 2019, the company released TensorFlow 2, an overhaul expanding its functionality while also fixing its flaws (TensorFlow Team, 2019). Indeed, writer Ray Johns (2021) observes that "the TensorFlow team [adopted] many of PyTorch's most popular features."

Johns particularly noted that eager execution, or the immediate evaluation of operations without building graphs, "inspired TensorFlow 2.0" (2021). Not only does this mean graphs no longer need to be defined statically, but it also dramatically simplified the coding API. To this point, Johns observed that "the APIs for both [frameworks now] look a lot alike." Similarly, Jocopo Mangiavacchi (2021), Senior Data Scientist at Microsoft, concluded that the "APIs are very similar" and "gave us very similar results" when training two simple models and using auto-differentiation and gradient descent.

Further, TensorFlow 2.0 improved parallelism and added multi-GPU support—a previous PyTorch advantage—delivering up to 3x faster training performance (TensorFlow Team, 2019).

But Google did not simply play catchup to Facebook with its 2.0 release. TensorFlow also added support for numerous programming languages and contexts PyTorch does not support. As its name implies, PyTorch was built especially for Python, though it has less-comprehensive interfaces for C++ and Julia. On the other hand, TensorFlow supports these languages but also C, Java, Go, JavaScript, R, and Swift (TensorFlow Team, 2019).

While Python may be one of the most heavily used languages by researchers due its high volume of robust NLP and AI frameworks, a variety of commercial applications are written in C and Java, client-side web scenarios require support of JavaScript, and iOS applications requires support of Swift. Instead, PyTorch applications must be wrapped in a RESTful web API where ML operations are performed on a remote, back-end Python server (Dubovikov, 2017).

However, TensorFlow is still slower than PyTorch and only support NVIDIA GPUs. One company found in a performance benchmark of image processing across many models that TensorFlow 2 to be only about 72% the speed of PyTorch—though accuracy was identical and TensorFlow utilized less than half the memory (Boesch, 2021). Another point TensorFlow is that although 2.x improved its native debugging capabilities, it still relies heavily on its own propriety debugging and visualization tools. On the other hand, PyTorch's limited visualization capabilities when building graph data has led many PyTorch users to connect it to its competitor's TensorBoard visualization tool (Boesch, 2021).

Conclusion

Overall, TensorFlow seems to have caught up to, and in many areas surpassed, PyTorch in all the areas that had previously caused PyTorch to overtake the research community. Perhaps most telling is that PyTorch Principle Engineer, Soumith Chintala (2020), tweeted after the release of TensorFlow 2 that, "debates on PyTorch vs TensorFlow were fun in 2017. […] Now both products look exactly the same, [and] the debates are nonsense."

Both libraries are open-sourced yet primarily developed by their respective creator companies. The difference is that, whereas Facebook is primarily a social media company, Google is, as one columnist noted, "no longer a search company—it's a machine learning company" (Oremus, 2015). PyTorch's reputation is that it's more commonly used in R&D than production (Johns, 2021).

These two perceptions, combined with TensorFlow's more extensive language support and distribution capabilities, probably give TensorFlow more of an advantage for production use outside of research. Indeed, in its 2021 survey of 41.7K professional developers, Stack Overflow found that TensorFlow was 1.7x more popular than PyTorch for the prior 12 months.

References

Boesch, G. (2021, March 2). *Pytorch vs Tensorflow: A Head-to-Head Comparison*. Deep Learning. Retreived November 4, 2021, from https://viso.ai/deep-learning/pytorch-vs-tensorflow.

Chintala, S. (2020, May 22). *Debates on PyTorch vs TensorFlow were fun in 2017. There was healthy competition to innovate, and* philosophical *differences like Theano vs Torch, Emacs vs vim, or android vs iOS. Now both products look exactly the same, the debates are nonsense and boring. Please stop*. Twitter. Retreived November 3, 2021, from https://twitter.com/soumithchintala/status/1263854044289929221.

Dean, J; Monga, R; et al. (2015, November 9). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. Google Research. Retrieved November 5, 2021, from http://download.tensorflow.org/paper/whitepaper2015.pdf.

Dubovikov, K. (2017, June 20). *PyTorch vs Tensorflow – spotting the difference*. Towards Data Science. Retrieved November 5, 2021, from https://towardsdatascience.com/pytorch-vs-tensorflow-spotting-the-difference-25c75777377b.

Gavigan, D. (2018, April 2). *The History of Angular*. Medium. Retrieved November 4, 2021, from https://medium.com/the-startup-lab-blog/the-history-of-angular-3e36f7e828c7.

He, H. (2019, October 10). *The State of Machine Learning Frameworks in 2019*. The Gradient. Retrieved November 1, 2021, from https://thegradient.pub/state-of-ml-frameworks-2019-pytorch-dominates-research-tensorflow-dominates-industry.

Johns, R. (2021, June 26). *PyTorch vs TensorFlow for Your Python Deep Learning Project*. Real Python. Retrieved November 5, 2021, from https://realpython.com/pytorch-vs-tensorflow.

Mangiavacchi, Jacopo. (2021, February 2). *A tale of two frameworks: PyTorch vs. TensorFlow*. Data Science at Microsoft. Retrieved November 6, 2021, from https://medium.com/data-science-at-microsoft/a-tale-of-two-frameworks-pytorch-vs-tensorflow-f73a975e733d.

Oremus, W. (2015, November 9). *What Is "TensorFlow," and Why Is Google So Excited About It?*. Slate. Retrieved November 5, 2021, from https://slate.com/technology/2015/11/google-s-tensorflow-is-open-source-and-it-s-about-to-be-a-huge-huge-deal.html.

Pichai, S. (2015, November 9). *TensorFlow: smarter machine learning, for everyone*. Google AI Blog. Retrieved November 3, 2021, from https://blog.google/technology/ai/tensorflow-smarter-machine-learning-for.

Stack Overflow. (2021, May). *2021 Developer Survey*. Stack Overflow. Retrieved November 5, 2021, from https://insights.stackoverflow.com/survey/2021#most-popular-technologies-misc-tech-prof.

TensorFlow Team. (2019, September 30). *TensorFlow 2.0 is now available!*. TensorFlow Blog. Retrieved November 3, 2021, from https://blog.tensorflow.org/2019/09/tensorflow-20-is-now-available.html.

Yegulalp, S. (2017, January 19). *Facebook brings GPU-powered machine learning to Python*. InfoWorld Tech Watch. Retrieved November 5, 2021, from https://www.infoworld.com/article/3159120/facebook-brings-gpu-powered-machine-learning-to-python.html.