

<https://towardsdatascience.com/pytorch-vs-tensorflow-1-month-summary-35d138590f9>

Dominic Monn

PyTorch vs. TensorFlow: 1 month summary



[Dominic Monn](#)

Follow

[Oct 8, 2017](#) · 3 min read

How PyTorch compares to TensorFlow after one month of working with PyTorch.





I've been a TensorFlow user for the better part of my Deep Learning work. However, when I joined [NVIDIA](#), we decided to switch to PyTorch — just as a test. These are my experiences with it.

Installation

The installation is very easy and straightforward. PyTorch can be installed via PIP or can be built from source. PyTorch also offers Docker images which can be used as a base image for your own project.

There isn't a designated CPU and GPU version of PyTorch like there is with TensorFlow. While this makes installation easier, it

generates more code if you want to support both, CPU and GPU usage.

It is to note that PyTorch does not offer an official windows distribution yet. There are non-official ports to windows, but there is no support from PyTorch.

Usage

PyTorch offers a very Pythonic API. This is very different from TensorFlow, where you are supposed to define all Tensors and the Graph and then run it in a session.

In my opinion, this leads to more, but much cleaner code. PyTorch Graphs have to be defined in a class which inherits from the `PyTorch nn.Module` class. A `forward()` function gets called when the Graph is run. With this "convention over configuration" approach the location of the graph is always known and variables aren't defined all over in the rest of the code.

This "new" approach needs some time to get used to, but I think it is very intuitive if you have worked with Python outside of Deep Learning before.

Based on some reviews, PyTorch also shows a better performance on a lot of models compared to TensorFlow.

Documentation

Documentation is complete for the most part. I never failed to find the definition of a function or module. Opposed to TensorFlow, where all functions have a single page, PyTorch only uses one page

per module. This makes it a bit more difficult if you are coming from Google, looking for a function.

Community

Obviously the community of PyTorch isn't as large as the one of TensorFlow. However, many people enjoy working with PyTorch in their free time, even though they use TensorFlow for work. I think this could change as soon as PyTorch gets out of Beta.

At the current moment, it is still a bit more difficult to find proficient people in PyTorch.

The community is large enough that questions in the official forums usually get a quick answer and so that a lot of example implementations of great neural networks got translated into PyTorch.

Tools and helpers

Even though PyTorch offers a fair amount of tools, some very useful ones are missing. One of the most helpful tools that are missing is TensorFlow's TensorBoard. This makes visualization a bit more difficult.

There are also some very common used helpers missing. This requires a bit more self-written code than TensorFlow.

Conclusion

PyTorch is an awesome alternative to TensorFlow. Since PyTorch is still in Beta, I expect some more changes and improvements to the usability, docs and performance.

PyTorch is very pythonic and feels comfortable to work with. It has a good community and documentation. It is also said to be a bit faster than TensorFlow.

However, the community is still quite smaller as opposed to TensorFlow and some useful tools such as the TensorBoard are missing.