

# Which Deep Learning Framework is Growing Fastest?

TensorFlow vs. PyTorch



Jeff Hale [Follow](#)

Apr 1 · 8 min read ★

In September 2018, I compared all the major deep learning frameworks in terms of demand, usage, and popularity in this article. TensorFlow was the undisputed heavyweight champion of deep learning frameworks. PyTorch was the young rookie with lots of buzz.



How has the landscape changed for the leading deep learning frameworks in the past six months?



To answer that question, I looked at the number of job listings on Indeed, Monster, LinkedIn, and SimplyHired. I also evaluated changes in Google search volume, GitHub activity, Medium articles, ArXiv articles, and Quora topic followers. Overall, these sources paint a comprehensive picture of growth in demand, usage, and interest.

## Integrations and Updates

We've recently seen several important developments in the TensorFlow and PyTorch frameworks.

PyTorch v1.0 was pre-released in October 2018, at the same time fastai v1.0 was released. Both releases marked major milestones in the maturity of the frameworks.

TensorFlow 2.0 alpha was released March 4, 2019. It added new features and an improved user experience. It more tightly integrates Keras as its high-level API, too.

## Methodology

In this article, I include Keras and fastai in the comparisons because of their tight integrations with TensorFlow and PyTorch. They also provide scale for evaluating TensorFlow and PyTorch.



I won't be exploring other deep learning frameworks in this article. I expect I will receive feedback that Caffe, Theano, MXNET, CNTK, DeepLearning4J, or Chainer deserve to be discussed. While these frameworks each have their virtues, none appear to be on a growth trajectory likely to put them near TensorFlow or PyTorch. Nor are they tightly coupled with either of those frameworks.

Searches were performed on March 20–21, 2019. Source data is in this [Google Sheet](#).

I used the plotly data visualization library to explore popularity. For the interactive plotly charts, see my [Kaggle Kernel](#) here.

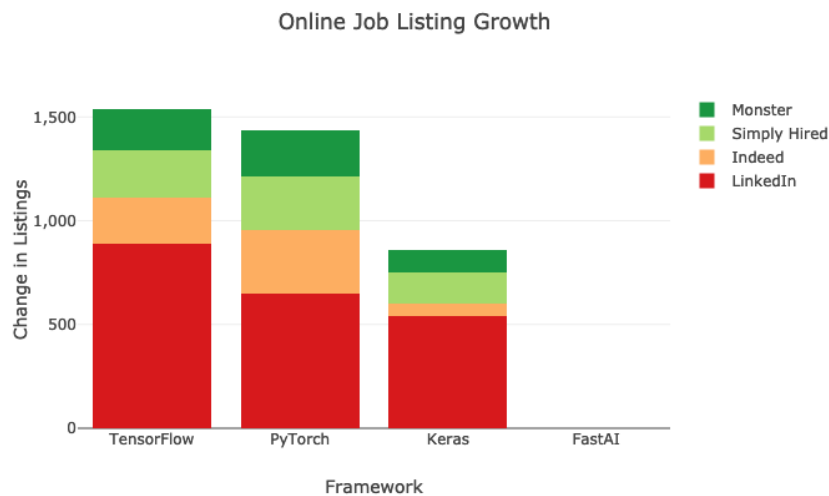
Let's look at the results in each category.

# Change in Online Job Listings

To determine which deep learning libraries are in demand in today's job market I searched job listings on Indeed, LinkedIn, Monster, and SimplyHired.

I searched with the term *machine learning*, followed by the library name. So TensorFlow was evaluated with *machine learning TensorFlow*. This method was used for historical comparison reasons. Searching without *machine learning* didn't yield appreciably different results. The search region was the USA.

I subtracted the number of listings six months ago from the number of listings in March 2019. Here's what I found:



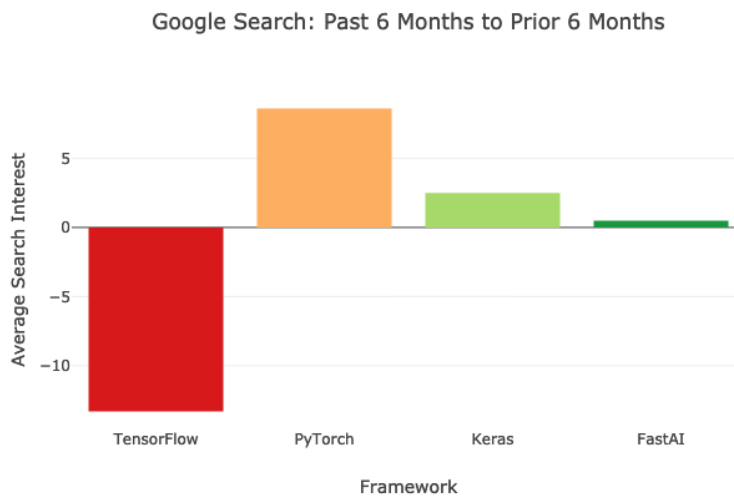
TensorFlow had a slightly larger increase in listings than PyTorch. Keras also saw listings growth—about half as much as TensorFlow. Fastai still isn't showing in hardly any job listings.

Note that PyTorch saw a larger number of additional listings than TensorFlow on all job search sites other than LinkedIn. Also note that in absolute terms, TensorFlow appears in nearly three times the number of job listings as PyTorch or Keras.

# Change in Average Google Search Activity

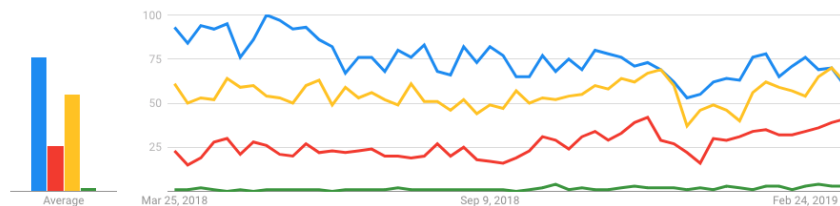
Web searches on the largest search engine are a gauge of popularity. I looked at search history in Google Trends over the past year. I searched for worldwide interest in the *Machine Learning and Artificial Intelligence* category. Google doesn't provide absolute search numbers, but it does provide relative figures.

I took the average interest score of the past six months and compared it to the average interest score for the prior six months.



In the past six months, the relative search volume for TensorFlow has decreased, while the relative search volume for PyTorch has grown.

The chart from Google directly below shows search interest over the past year.

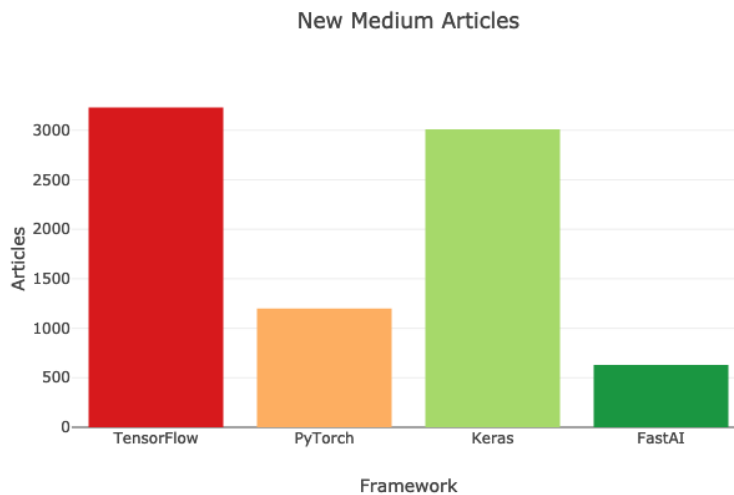


TensorFlow in blue; Keras in yellow, PyTorch in red, fastai in green

## New Medium Articles

Medium is a popular location for data science articles and tutorials. I hope you're enjoying it! 😊

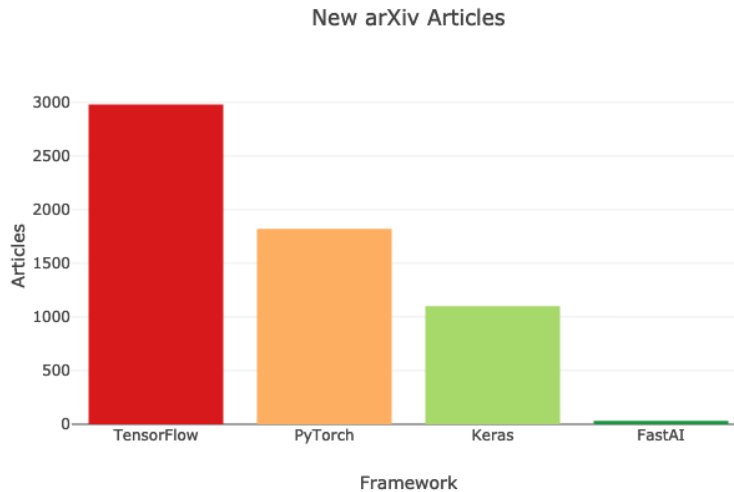
I used Google site search of Medium.com over the past six months and found TensorFlow and Keras had similar numbers of articles published. PyTorch had relatively few.



As high level APIs, Keras and fastai are popular with new deep learning practitioners. Medium has many tutorials showing how to use these frameworks.

## New arXiv Articles

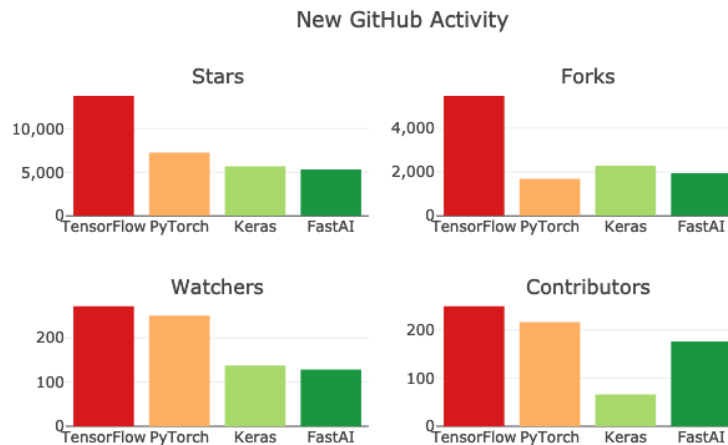
arXiv is the online repository where most scholarly deep learning articles are published. I searched for new articles mentioning each framework on arXiv using Google site search results for the past six months.



TensorFlow had the most new article appearances by a good margin.

## New GitHub Activity

Recent activity on GitHub is another indicator of framework popularity. I broke out stars, forks, watchers, and contributors in the charts below.

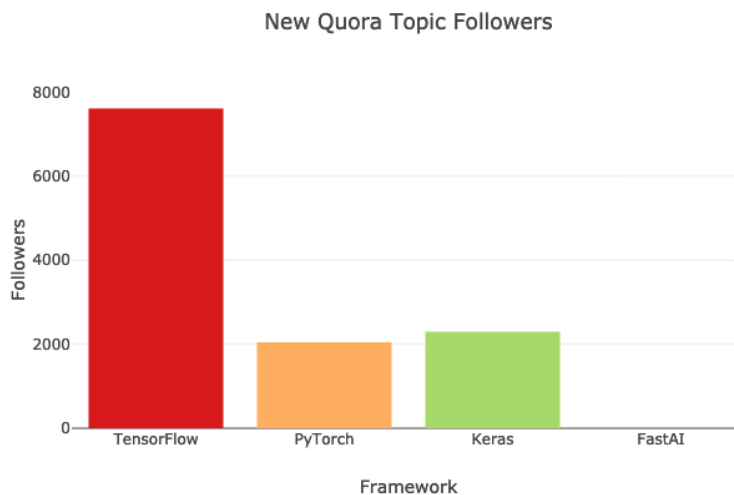


TensorFlow had the most GitHub activity in each category. However, PyTorch was quite close in terms of growth in watchers and contributors. Also, Fastai saw many new contributors.

Some contributors to Keras are no doubt working on it in the TensorFlow library. It's worth noting that both TensorFlow and Keras are open source products spearheaded by Googlers.

## New Quora Followers

I added the number of new Quora topic followers to the mix—a new category that I didn't have the data for previously.



TensorFlow added the most new topic followers over the past six months. PyTorch and Keras each added far fewer.

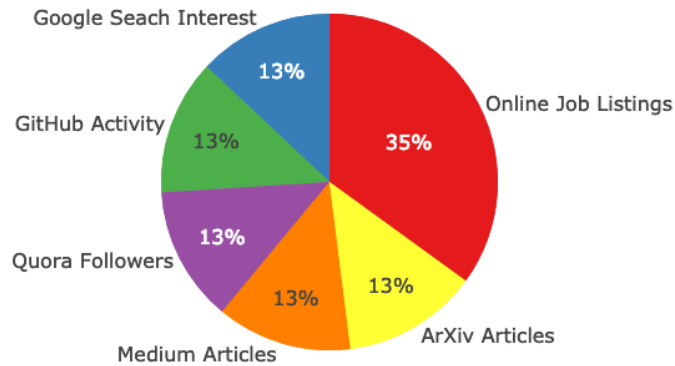
Once I had all the data, I consolidated it into one metric.

## Growth Score Procedure

Here's how I created the growth score:

1. Scaled all features between 0 and 1.
2. Aggregated the *Online Job Listings* and *GitHub Activity* subcategories.
3. Weighted categories according to the percentages below.

Weights by Category



4. Multiplied weighted scores by 100 for comprehensibility.

5. Summed category scores for each framework into a single growth score.

Job listings make up a little over a third of the total score. As the cliché goes, money talks. 🤑 This split seemed like an appropriate balance of the various categories. Unlike my 2018 power score analysis, I didn't include KDNuggets usage survey (no new data) or books (not many published in six months).

## Results

Here are the changes in tabular form.

Framework	Indeed	Monster	Simply_Hired	LinkedIn	Google	Medium	ArXiv	Quora	Stars	Watchers	Forks	Contributors
TensorFlow	222	204	224	891	-13.3	3,230	2,980	7,616	13,775	271	5,466	249
PyTorch	308	225	257	647	8.6	1,200	1,820	2,049	7,267	250	1,684	216
Keras	58	107	154	542	2.5	3,010	1,100	2,303	5,697	137	2,294	66
FastAI	1	1	1	0	0.5	631	31	0	5,341	128	1,943	176

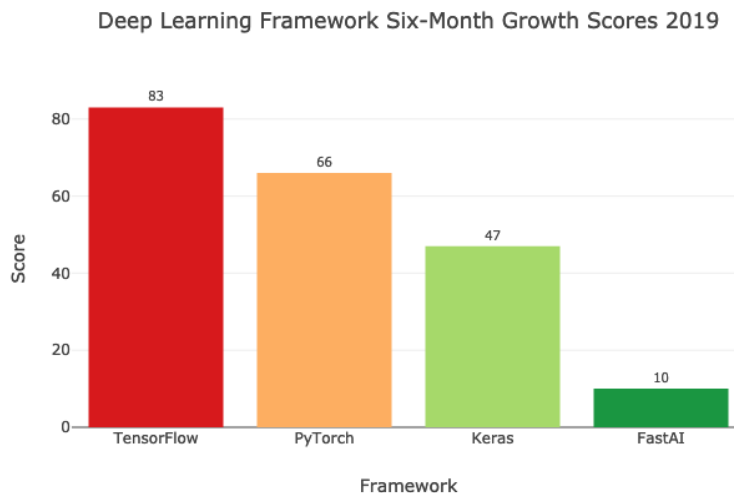
[Google Sheet here.](#)

Here are the category and final scores.



	Job Search Listings	Google Interest	GitHub Activity	ArXiv Articles	Medium Articles	Quora Followers	Growth Score
<b>Framework</b>							
<b>TensorFlow</b>	31.00	0.00	13.00	13.00	13.00	13.00	83.00
<b>PyTorch</b>	33.00	13.00	6.00	8.00	3.00	3.00	66.00
<b>Keras</b>	16.00	9.00	1.00	5.00	12.00	4.00	47.00
<b>FastAI</b>	0.00	8.00	2.00	0.00	0.00	0.00	10.00

Here are the final growth scores.



TensorFlow is both the most in demand framework and the fastest growing. It's not going anywhere anytime soon. 🤔 PyTorch is growing rapidly, too. Its large increase in job listings is evidence of its increased usage and demand. Keras has grown a good bit in the past six months, also. Finally, fastai has grown from a low baseline. It's worth remembering that it's the youngest of the lot.

Both TensorFlow and PyTorch are good frameworks to learn.

## Learning Suggestions

If you're looking to learn TensorFlow, I suggest you start with Keras. I recommend Chollet's *Deep Learning with Python* and Dan Becker's DataCamp course on Keras. Tensorflow 2.0 is using Keras as its high-level API through `tf.keras`. Here's a quick getting started intro to TensorFlow 2.0 by Chollet.

If you're looking to learn PyTorch, I suggest you start with fast.ai's MOOC Practical Deep Learning for Coders, v3. You'll learn deep learning fundamentals, fastai, and PyTorch basics.

What's ahead for TensorFlow and PyTorch?

## Future Directions

I've consistently heard that folks enjoy using PyTorch more than TensorFlow. PyTorch is more pythonic and has a more consistent API. It also has native ONNX model exports, which can be used to speed up inference. Also, PyTorch shares many commands with numpy, which reduces the barrier to learning it.

However, TensorFlow 2.0 is all about improved UX, as Google's Chief Decision Intelligence Engineer, [Cassie Kozyrkov](#), explains here. TensorFlow will now have a more straightforward API, a streamlined Keras integration, and an eager execution option. These changes, and TensorFlow's broad adoption, should help the framework remain popular for years to come.

TensorFlow recently announced another exciting plan: the development of Swift for TensorFlow. Swift is a programming language originally built by Apple. Swift has a number of advantages over Python in terms of execution and development speed. Fast.ai will be using Swift for TensorFlow in part of its advanced MOOC—see fast.ai cofounder Jeremy Howard's post on the topic here. The language probably won't be ready for prime time for a year or two, but it could be an improvement over current deep learning frameworks.

Collaboration and cross-pollination among languages and frameworks is definitely happening. 🐝 🌸

Another advancement that will affect deep learning frameworks is quantum computing. A usable quantum computer is likely a few years off, but Google, IBM, Microsoft, and others are thinking about how to integrate quantum computing with deep learning. Frameworks will need to be adapted to work with this new technology.

# Wrap

You've seen that both TensorFlow and PyTorch are growing. Both now have nice high-level APIs—`tf.keras` and `fastai`—that have lowered the barriers to getting started with deep learning. You've also heard a bit about recent developments and future directions.

To play with the charts in this article interactively or fork the Jupyter Notebook, please head to my Kaggle Kernel.

I hope you've found this comparison helpful. If you have, please share it on your favorite social media channel so others can find it, too. 😊

I write about deep learning, DevOps, data science, and other tech topics. If any of those are of interest to you, check them out and follow me here.

To make sure you don't miss out on great content, join my mailing list below.

**Join my [Data Awesome](#) mailing list. One email per month of awesome curated content!**

**Email Address**

Thanks for reading!



