

# TensorFlow

**TensorFlow** is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.<sup>[4][5]</sup>

TensorFlow was developed by the Google Brain team for internal Google use in research and production.<sup>[6][7][8]</sup> The initial version was released under the Apache License 2.0 in 2015.<sup>[1][9]</sup> Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019.<sup>[10]</sup>

TensorFlow can be used in a wide variety of programming languages, including Python, JavaScript, C++, and Java.<sup>[11]</sup> This flexibility lends itself to a range of applications in many different sectors.

## History

### DistBelief

Starting in 2011, Google Brain built DistBelief as a proprietary machine learning system based on deep learning neural networks. Its use grew rapidly across diverse Alphabet companies in both research and commercial applications.<sup>[12][13]</sup> Google assigned multiple computer scientists, including Jeff Dean, to simplify and refactor the codebase of DistBelief into a faster, more robust application-grade library, which became TensorFlow.<sup>[14]</sup> In 2009, the team, led by Geoffrey Hinton, had implemented generalized backpropagation and other improvements which allowed generation of neural networks with substantially higher accuracy, for instance a 25% reduction in errors in speech recognition.<sup>[15]</sup>

### TensorFlow

TensorFlow is Google Brain's second-generation system. Version 1.0.0 was released on February 11, 2017.<sup>[16]</sup> While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units).<sup>[17]</sup> TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

### TensorFlow

 <b>TensorFlow</b>	
<b>Developer(s)</b>	<u>Google Brain Team</u> <sup>[1]</sup>
<b>Initial release</b>	November 9, 2015
<b>Stable release</b>	2.11.0 <sup>[2]</sup>  (18 November 2022)
<b>Repository</b>	<u>github.com</u> <u>/tensorflow</u> <u>/tensorflow</u> ( <a href="https://github.com/tensorflow/tensorflow">https://github.com/tensorflow/tensorflow</a> )
<b>Written in</b>	<u>Python</u> , <u>C++</u> , <u>CUDA</u>
<b>Platform</b>	<u>Linux</u> , <u>macOS</u> , <u>Windows</u> , <u>Android</u> , <u>JavaScript</u> <sup>[3]</sup>
<b>Type</b>	<u>Machine learning library</u>
<b>License</b>	<u>Apache License 2.0</u>
<b>Website</b>	<u>www.tensorflow.org</u> ( <a href="https://www.tensorflow.org">https://www.tensorflow.org</a> )

TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors. During the Google I/O Conference in June 2016, Jeff Dean stated that 1,500 repositories on GitHub mentioned TensorFlow, of which only 5 were from Google.<sup>[18]</sup>

In December 2017, developers from Google, Cisco, RedHat, CoreOS, and CaiCloud introduced Kubeflow at a conference. Kubeflow allows operation and deployment of TensorFlow on Kubernetes.

In March 2018, Google announced TensorFlow.js version 1.0 for machine learning in JavaScript.<sup>[19]</sup>

In Jan 2019, Google announced TensorFlow 2.0.<sup>[20]</sup> It became officially available in Sep 2019.<sup>[10]</sup>

In May 2019, Google announced TensorFlow Graphics for deep learning in computer graphics.<sup>[21]</sup>

## Tensor processing unit (TPU)

In May 2016, Google announced its Tensor processing unit (TPU), an application-specific integrated circuit (ASIC, a hardware chip) built specifically for machine learning and tailored for TensorFlow. A TPU is a programmable AI accelerator designed to provide high throughput of low-precision arithmetic (e.g., 8-bit), and oriented toward using or running models rather than training them. Google announced they had been running TPUs inside their data centers for more than a year, and had found them to deliver an order of magnitude better-optimized performance per watt for machine learning.<sup>[22]</sup>

In May 2017, Google announced the second-generation, as well as the availability of the TPUs in Google Compute Engine.<sup>[23]</sup> The second-generation TPUs deliver up to 180 teraflops of performance, and when organized into clusters of 64 TPUs, provide up to 11.5 petaflops.

In May 2018, Google announced the third-generation TPUs delivering up to 420 teraflops of performance and 128 GB high bandwidth memory (HBM). Cloud TPU v3 Pods offer 100+ petaflops of performance and 32 TB HBM.<sup>[24]</sup>

In February 2018, Google announced that they were making TPUs available in beta on the Google Cloud Platform.<sup>[25]</sup>

## Edge TPU

In July 2018, the Edge TPU was announced. Edge TPU is Google's purpose-built ASIC chip designed to run TensorFlow Lite machine learning (ML) models on small client computing devices such as smartphones<sup>[26]</sup> known as edge computing.

## TensorFlow Lite

In May 2017, Google announced a software stack specifically for mobile development, TensorFlow Lite.<sup>[27]</sup> In January 2019, the TensorFlow team released a developer preview of the mobile GPU inference engine with OpenGL ES 3.1 Compute Shaders on Android devices and Metal Compute Shaders on iOS devices.<sup>[28]</sup> In May 2019, Google announced that their TensorFlow Lite Micro (also known as TensorFlow Lite for Microcontrollers) and ARM's uTensor would be merging.<sup>[29]</sup>

## Pixel Visual Core (PVC)

In October 2017, Google released the Google Pixel 2 which featured their Pixel Visual Core (PVC), a fully programmable image, vision and AI processor for mobile devices. The PVC supports TensorFlow for machine learning (and Halide for image processing).

## TensorFlow 2.0

As TensorFlow's market share among research papers was declining to the advantage of PyTorch,<sup>[30]</sup> the TensorFlow Team announced a release of a new major version of the library in September 2019. TensorFlow 2.0 introduced many changes, the most significant being TensorFlow eager, which changed the automatic differentiation scheme from the static computational graph, to the "Define-by-Run" scheme originally made popular by Chainer and later PyTorch.<sup>[30]</sup> Other major changes included removal of old libraries, cross-compatibility between trained models on different versions of TensorFlow, and significant improvements to the performance on GPU.<sup>[31]</sup>

## Features

---

### AutoDifferentiation

AutoDifferentiation is the process of automatically calculating the gradient vector of a model with respect to each of its parameters. With this feature, TensorFlow can automatically compute the gradients for the parameters in a model, which is useful to algorithms such as backpropagation which require gradients to optimize performance.<sup>[32]</sup> To do so, the framework must keep track of the order of operations done to the input Tensors in a model, and then compute the gradients with respect to the appropriate parameters.<sup>[32]</sup>

### Eager execution

TensorFlow includes an “eager execution” mode, which means that operations are evaluated immediately as opposed to being added to a computational graph which is executed later.<sup>[33]</sup> Code executed eagerly can be examined step-by-step through a debugger, since data is augmented at each line of code rather than later in a computational graph.<sup>[33]</sup> This execution paradigm is considered to be easier to debug because of its step by step transparency.<sup>[33]</sup>

### Distribute

In both eager and graph executions, TensorFlow provides an API for distributing computation across multiple devices with various distribution strategies.<sup>[34]</sup> This distributed computing can often speed up the execution of training and evaluating of TensorFlow models and is a common practice in the field of AI.<sup>[34][35]</sup>

## Losses

To train and assess models, TensorFlow provides a set of loss functions (also known as cost functions).<sup>[36]</sup> Some popular examples include mean squared error (MSE) and binary cross entropy (BCE).<sup>[36]</sup> These loss functions compute the “error” or “difference” between a model's output and the expected output (more broadly, the difference between two tensors). For different datasets and models, different losses are used to prioritize certain aspects of performance.

## Metrics

In order to assess the performance of machine learning models, TensorFlow gives API access to commonly used metrics. Examples include various accuracy metrics (binary, categorical, sparse categorical) along with other metrics such as Precision, Recall, and Intersection-over-Union (IoU).<sup>[37]</sup>

## TF.nn

TensorFlow.nn is a module for executing primitive neural network operations on models.<sup>[38]</sup> Some of these operations include variations of convolutions (1/2/3D, Atrous, depthwise), activation functions (Softmax, RELU, GELU, Sigmoid, etc.) and their variations, and other Tensor operations (max-pooling, bias-add, etc.).<sup>[38]</sup>

## Optimizers

TensorFlow offers a set of optimizers for training neural networks, including ADAM, ADAGRAD, and Stochastic Gradient Descent (SGD).<sup>[39]</sup> When training a model, different optimizers offer different modes of parameter tuning, often affecting a model's convergence and performance.<sup>[40]</sup>

## Usage and extensions

---

### TensorFlow

TensorFlow serves as a core platform and library for machine learning. TensorFlow's APIs use Keras to allow users to make their own machine learning models.<sup>[41]</sup> In addition to building and training their model, TensorFlow can also help load the data to train the model, and deploy it using TensorFlow Serving.<sup>[42]</sup>

TensorFlow provides a stable Python API,<sup>[43]</sup> as well as APIs without backwards compatibility guarantee for Javascript,<sup>[44]</sup> C++,<sup>[45]</sup> and Java.<sup>[46][11]</sup> Third-party language binding packages are also available for C#,<sup>[47][48]</sup> Haskell,<sup>[49]</sup> Julia,<sup>[50]</sup> MATLAB,<sup>[51]</sup> Object Pascal,<sup>[52]</sup> R,<sup>[53]</sup> Scala,<sup>[54]</sup> Rust,<sup>[55]</sup> OCaml,<sup>[56]</sup> and Crystal.<sup>[57]</sup> Bindings that are now archived and unsupported include Go<sup>[58]</sup> and Swift.<sup>[59]</sup>

## TensorFlow.js

TensorFlow also has a library for machine learning in JavaScript. Using the provided JavaScript APIs, TensorFlow.js allows users to use either Tensorflow.js models or converted models from TensorFlow or TFLite, retrain the given models, and run on the web.<sup>[42][60]</sup>

## TFLite

TensorFlow Lite has APIs for mobile apps or embedded devices to generate and deploy TensorFlow models.<sup>[61]</sup> These models are compressed and optimized in order to be more efficient and have a higher performance on smaller capacity devices.<sup>[62]</sup>

TensorFlow Lite uses FlatBuffers as the data serialization format for network models, eschewing the Protocol Buffers format used by standard TensorFlow models.<sup>[62]</sup>

## TFX

TensorFlow Extended (abbrev. TFX) provides numerous components to perform all the operations needed for end-to-end production.<sup>[63]</sup> Components include loading, validating, and transforming data, tuning, training, and evaluating the machine learning model, and pushing the model itself into production.<sup>[42][63]</sup>

## Integrations

### Numpy

Numpy is one of the most popular Python data libraries, and TensorFlow offers integration and compatibility with its data structures.<sup>[64]</sup> Numpy NDarrays, the library's native datatype, are automatically converted to TensorFlow Tensors in TF operations; the same is also true vice versa.<sup>[64]</sup> This allows for the two libraries to work in unison without requiring the user to write explicit data conversions. Moreover, the integration extends to memory optimization by having TF Tensors share the underlying memory representations of Numpy NDarrays whenever possible.<sup>[64]</sup>

## Extensions

TensorFlow also offers a variety of libraries and extensions to advance and extend the models and methods used.<sup>[65]</sup> For example, TensorFlow Recommenders and TensorFlow Graphics are libraries for their respective functionalities in recommendation systems and graphics, TensorFlow Federated provides a framework for decentralized data, and TensorFlow Cloud allows users to directly interact with Google

Cloud to integrate their local code to Google Cloud.<sup>[66]</sup> Other add-ons, libraries, and frameworks include TensorFlow Model Optimization, TensorFlow Probability, TensorFlow Quantum, and TensorFlow Decision Forests.<sup>[65][66]</sup>

## Google Colab

Google also released Colaboratory, a TensorFlow Jupyter notebook environment that does not require any setup.<sup>[67]</sup> It runs on Google Cloud and allows users free access to GPUs and the ability to store and share notebooks on Google Drive.<sup>[68]</sup>

## Google JAX

Google JAX is a machine learning framework for transforming numerical functions.<sup>[69][70][71]</sup> It is described as bringing together a modified version of autograd (<https://github.com/HIPS/autograd>) (automatic obtaining of the gradient function through differentiation of a function) and TensorFlow's XLA (<https://www.tensorflow.org/xla>) (Accelerated Linear Algebra). It is designed to follow the structure and workflow of NumPy as closely as possible and works with TensorFlow as well as other frameworks such as PyTorch. The primary functions of JAX are:<sup>[69]</sup>

1. grad: automatic differentiation
2. jit: compilation
3. vmap: auto-vectorization
4. pmap: SPMD programming

# Applications

---

## Medical

GE Healthcare used TensorFlow to increase the speed and accuracy of MRIs in identifying specific body parts.<sup>[72]</sup> Google used TensorFlow to create DermAssist, a free mobile application that allows users to take pictures of their skin and identify potential health complications.<sup>[73]</sup> Sinovation Ventures used TensorFlow to identify and classify eye diseases from optical coherence tomography (OCT) scans.<sup>[73]</sup>

## Social media

Twitter implemented TensorFlow to rank tweets by importance for a given user, and changed their platform to show tweets in order of this ranking.<sup>[74]</sup> Previously, tweets were simply shown in reverse chronological order.<sup>[74]</sup> The photo sharing app VSCO used TensorFlow to help suggest custom filters for photos.<sup>[73]</sup>

## Search Engine

Google officially released RankBrain on October 26, 2015, backed by TensorFlow.<sup>[75]</sup>

## Education

InSpace, a virtual learning platform, used TensorFlow to filter out toxic chat messages in classrooms.<sup>[76]</sup> Liulishuo, an online English learning platform, utilized TensorFlow to create an adaptive curriculum for each student.<sup>[77]</sup> TensorFlow was used to accurately assess a student's current abilities, and also helped decide the best future content to show based on those capabilities.<sup>[77]</sup>

## Retail

The e-commerce platform Carousell used TensorFlow to provide personalized recommendations for customers.<sup>[73]</sup> The cosmetics company ModiFace used TensorFlow to create an augmented reality experience for customers to test various shades of make-up on their face.<sup>[78]</sup>

## Research



2016 comparison of original photo (left) and with TensorFlow *neural style* applied (right)

TensorFlow is the foundation for the automated image-captioning software DeepDream.<sup>[79]</sup>

## See also

---



***Free and open-  
source software  
portal***

- Comparison of deep learning software
- Differentiable programming
- Keras

## Bibliography

---

- Moroney, Laurence (October 1, 2020). *AI and Machine Learning for Coders* (<https://www.oreilly.com/library/view/ai-and-machine/9781492078180/>) (1st ed.). O'Reilly Media. p. 365. ISBN 9781492078197.
- Géron, Aurélien (October 15, 2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (<https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/>) (2nd ed.). O'Reilly Media. p. 856. ISBN 9781492032632.
- Ramsundar, Bharath; Zadeh, Reza Bosagh (March 23, 2018). *TensorFlow for Deep Learning* (<https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/>) (1st ed.). O'Reilly Media. p. 256. ISBN 9781491980446.
- Hope, Tom; Resheff, Yehezkel S.; Lieder, Itay (August 27, 2017). *Learning TensorFlow: A Guide to Building Deep Learning Systems* (<https://www.oreilly.com/library/view/learning-tensorflow/9781491978504/>) (1st ed.). O'Reilly Media. p. 242. ISBN 9781491978504.

- Shukla, Nishant (February 12, 2018). *Machine Learning with TensorFlow* (1st ed.). Manning Publications. p. 272. ISBN 9781617293870.

## External links

---

- Official website (<https://www.tensorflow.org>)
- Learning TensorFlow.js Book (ENG) (<https://www.oreilly.com/library/view/learning-tensorflow-js/9781492090786/>)

## References

---

1. "Credits" (<https://tensorflow.org/about>). *TensorFlow.org*. Retrieved November 10, 2015.
2. "Release 2.11.0" (<https://github.com/tensorflow/tensorflow/releases/tag/v2.11.0>). November 18, 2022. Retrieved November 27, 2022.
3. "TensorFlow.js" (<https://js.tensorflow.org/faq/>). Retrieved June 28, 2018.
4. Abadi, Martín; Barham, Paul; Chen, Jianmin; Chen, Zhifeng; Davis, Andy; Dean, Jeffrey; Devin, Matthieu; Ghemawat, Sanjay; Irving, Geoffrey; Isard, Michael; Kudlur, Manjunath; Levenberg, Josh; Monga, Rajat; Moore, Sherry; Murray, Derek G.; Steiner, Benoit; Tucker, Paul; Vasudevan, Vijay; Warden, Pete; Wicke, Martin; Yu, Yuan; Zheng, Xiaoqiang (2016). *TensorFlow: A System for Large-Scale Machine Learning* (<https://www.usenix.org/system/file/s/conference/osdi16/osdi16-abadi.pdf>) (PDF). Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16). arXiv:1605.08695 (<https://arxiv.org/abs/1605.08695>).
5. *TensorFlow: Open source machine learning* ([https://www.youtube.com/watch?v=oZikw5k\\_2FM](https://www.youtube.com/watch?v=oZikw5k_2FM)). Google. 2015. Archived ([https://ghostarchive.org/varchive/youtube/20211111/oZikw5k\\_2FM](https://ghostarchive.org/varchive/youtube/20211111/oZikw5k_2FM)) from the original on November 11, 2021. "It is machine learning software being used for various kinds of perceptual and language understanding tasks" – Jeffrey Dean, minute 0:47 / 2:17 from YouTube clip
6. Video clip by Google about TensorFlow 2015 at minute 0:15/2:17
7. Video clip by Google about TensorFlow 2015 at minute 0:26/2:17
8. Dean et al 2015, p. 2
9. Metz, Cade (November 9, 2015). "Google Just Open Sourced TensorFlow, Its Artificial Intelligence Engine" (<https://www.wired.com/2015/11/google-open-sources-its-artificial-intelligence-engine/>). *Wired*. Retrieved November 10, 2015.
10. TensorFlow (September 30, 2019). "TensorFlow 2.0 is now available!" (<https://medium.com/tensorflow/tensorflow-2-0-is-now-available-57d706c2a9ab>). *Medium*. Retrieved November 24, 2019.
11. "API Documentation" ([https://www.tensorflow.org/api\\_docs/](https://www.tensorflow.org/api_docs/)). Retrieved June 27, 2018.
12. Dean, Jeff; Monga, Rajat; et al. (November 9, 2015). "TensorFlow: Large-scale machine learning on heterogeneous systems" (<http://download.tensorflow.org/paper/whitepaper2015.pdf>) (PDF). *TensorFlow.org*. Google Research. Retrieved November 10, 2015.
13. Perez, Sarah (November 9, 2015). "Google Open-Sources The Machine Learning Tech Behind Google Photos Search, Smart Reply And More" (<https://techcrunch.com/2015/11/09/google-open-sources-the-machine-learning-tech-behind-google-photos-search-smart-reply-and-more/>). *TechCrunch*. Retrieved November 11, 2015.
14. Oremus, Will (November 9, 2015). "What Is TensorFlow, and Why Is Google So Excited About It?" ([https://www.slate.com/blogs/future\\_tense/2015/11/09/google\\_s\\_tensorflow\\_is\\_open\\_source\\_and\\_it\\_s\\_about\\_to\\_be\\_a\\_huge\\_huge\\_deal.html](https://www.slate.com/blogs/future_tense/2015/11/09/google_s_tensorflow_is_open_source_and_it_s_about_to_be_a_huge_huge_deal.html)). *Slate*. Retrieved November 11, 2015.



15. Ward-Bailey, Jeff (November 25, 2015). "Google chairman: We're making 'real progress' on artificial intelligence" (<https://www.csmonitor.com/Technology/2015/0914/Google-chairman-We-re-making-real-progress-on-artificial-intelligence>). *CSMonitor*. Retrieved November 25, 2015.
16. TensorFlow Developers (2022). "Tensorflow Release 1.0.0" (<https://github.com/tensorflow/tensorflow/blob/07bb8ea2379bd459832b23951fb20ec47f3fdbd4/RELEASE.md>). *GitHub*. doi:10.5281/zenodo.4724125 (<https://doi.org/10.5281%2Fzenodo.4724125>).
17. Metz, Cade (November 10, 2015). "TensorFlow, Google's Open Source AI, Points to a Fast-Changing Hardware World" (<https://www.wired.com/2015/11/googles-open-source-ai-tensorflow-signals-fast-changing-hardware-world/>). *Wired*. Retrieved November 11, 2015.
18. Machine Learning: Google I/O 2016 Minute 07:30/44:44 (<https://www.youtube.com/watch?v=Rnm83GqgqPE>) accessdate=2016-06-05
19. TensorFlow (March 30, 2018). "Introducing TensorFlow.js: Machine Learning in Javascript" (<https://medium.com/tensorflow/introducing-tensorflow-js-machine-learning-in-javascript-bf3eab376db>). *Medium*. Retrieved May 24, 2019.
20. TensorFlow (January 14, 2019). "What's coming in TensorFlow 2.0" (<https://medium.com/tensorflow/whats-coming-in-tensorflow-2-0-d3663832e9b8>). *Medium*. Retrieved May 24, 2019.
21. TensorFlow (May 9, 2019). "Introducing TensorFlow Graphics: Computer Graphics Meets Deep Learning" (<https://medium.com/tensorflow/introducing-tensorflow-graphics-computer-graphics-meets-deep-learning-c8e3877b7668>). *Medium*. Retrieved May 24, 2019.
22. Jouppi, Norm. "Google supercharges machine learning tasks with TPU custom chip" (<https://cloudplatform.googleblog.com/2016/05/Google-supercharges-machine-learning-tasks-with-custom-chip.html>). *Google Cloud Platform Blog*. Retrieved May 19, 2016.
23. "Build and train machine learning models on our new Google Cloud TPUs" (<https://www.blog.google/topics/google-cloud/google-cloud-offer-tpus-machine-learning/>). *Google*. May 17, 2017. Retrieved May 18, 2017.
24. "Cloud TPU" (<https://cloud.google.com/tpu/>). *Google Cloud*. Retrieved May 24, 2019.
25. "Cloud TPU machine learning accelerators now available in beta" (<https://cloudplatform.googleblog.com/2018/02/Cloud-TPU-machine-learning-accelerators-now-available-in-beta.html>). *Google Cloud Platform Blog*. Retrieved February 12, 2018.
26. Kundu, Kishalaya (July 26, 2018). "Google Announces Edge TPU, Cloud IoT Edge at Cloud Next 2018" (<https://beebom.com/google-announces-edge-tpu-cloud-iot-edge-at-cloud-next-2018/>). *Beebom*. Retrieved February 2, 2019.
27. "Google's new machine learning framework is going to put more AI on your phone" (<https://www.theverge.com/2017/5/17/15645908/google-ai-tensorflowlite-machine-learning-announcement-io-2017>). May 17, 2017.
28. TensorFlow (January 16, 2019). "TensorFlow Lite Now Faster with Mobile GPUs (Developer Preview)" (<https://medium.com/tensorflow/tensorflow-lite-now-faster-with-mobile-gpus-developer-preview-e15797e6dee7>). *Medium*. Retrieved May 24, 2019.
29. "uTensor and Tensor Flow Announcement | Mbed" (<https://os.mbed.com/blog/entry/uTensor-and-Tensor-Flow-Announcement/>). *os.mbed.com*. Retrieved May 24, 2019.
30. He, Horace (October 10, 2019). "The State of Machine Learning Frameworks in 2019" (<https://thegradient.pub/state-of-ml-frameworks-2019-pytorch-dominates-research-tensorflow-dominates-industry/>). *The Gradient*. Retrieved May 22, 2020.
31. "TensorFlow 2.0 is now available!" (<https://blog.tensorflow.org/2019/09/tensorflow-20-is-now-available.html>). *TensorFlow Blog*. September 30, 2019. Retrieved May 22, 2020.
32. "Introduction to gradients and automatic differentiation" (<https://www.tensorflow.org/guide/autodiff>). *TensorFlow*. Retrieved November 4, 2021.

33. "Eager execution | TensorFlow Core" (<https://www.tensorflow.org/guide/eager>). *TensorFlow*. Retrieved November 4, 2021.
34. "Module: tf.distribute | TensorFlow Core v2.6.1" ([https://www.tensorflow.org/api\\_docs/python/tf/distribute](https://www.tensorflow.org/api_docs/python/tf/distribute)). *TensorFlow*. Retrieved November 4, 2021.
35. Sigeru., Omatu (2014). *Distributed Computing and Artificial Intelligence, 11th International Conference* (<http://worldcat.org/oclc/980886715>). Springer International Publishing. ISBN 978-3-319-07593-8. OCLC 980886715 (<https://www.worldcat.org/oclc/980886715>).
36. "Module: tf.losses | TensorFlow Core v2.6.1" ([https://www.tensorflow.org/api\\_docs/python/tf/losses](https://www.tensorflow.org/api_docs/python/tf/losses)). *TensorFlow*. Retrieved November 4, 2021.
37. "Module: tf.metrics | TensorFlow Core v2.6.1" ([https://www.tensorflow.org/api\\_docs/python/tf/metrics](https://www.tensorflow.org/api_docs/python/tf/metrics)). *TensorFlow*. Retrieved November 4, 2021.
38. "Module: tf.nn | TensorFlow Core v2.7.0" ([https://www.tensorflow.org/api\\_docs/python/tf/nn](https://www.tensorflow.org/api_docs/python/tf/nn)). *TensorFlow*. Retrieved November 6, 2021.
39. "Module: tf.optimizers | TensorFlow Core v2.7.0" ([https://www.tensorflow.org/api\\_docs/python/tf/optimizers](https://www.tensorflow.org/api_docs/python/tf/optimizers)). *TensorFlow*. Retrieved November 6, 2021.
40. Dogo, E. M.; Afolabi, O. J.; Nwulu, N. I.; Twala, B.; Aigbavboa, C. O. (December 2018). "A Comparative Analysis of Gradient Descent-Based Optimization Algorithms on Convolutional Neural Networks" ([https://ieeexplore.ieee.org/abstract/document/8769211?casa\\_token=uCDWT8\\_i8jYAAAAA:NvEsZNEoUIDWY3mCHS7syTys71el11x7wgAPH\\_t26BQqUZLY-mASNZvCwOAcYz6UBhxFBZvEITY](https://ieeexplore.ieee.org/abstract/document/8769211?casa_token=uCDWT8_i8jYAAAAA:NvEsZNEoUIDWY3mCHS7syTys71el11x7wgAPH_t26BQqUZLY-mASNZvCwOAcYz6UBhxFBZvEITY)). *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*: 92–99. doi:10.1109/CTEMS.2018.8769211 (<https://doi.org/10.1109%2FCTEMS.2018.8769211>). ISBN 978-1-5386-7709-4. S2CID 198931032 (<https://api.semanticscholar.org/CorpusID:198931032>).
41. "TensorFlow Core | Machine Learning for Beginners and Experts" (<https://www.tensorflow.org/overview>). *TensorFlow*. Retrieved November 4, 2021.
42. "Introduction to TensorFlow" (<https://www.tensorflow.org/learn>). *TensorFlow*. Retrieved October 28, 2021.
43. "All symbols in TensorFlow 2 | TensorFlow Core v2.7.0" ([https://www.tensorflow.org/api\\_docs/python/tf/all\\_symbols](https://www.tensorflow.org/api_docs/python/tf/all_symbols)). *TensorFlow*. Retrieved November 6, 2021.
44. "TensorFlow.js" (<https://js.tensorflow.org/>). *js.tensorflow.org*. Retrieved November 6, 2021.
45. "TensorFlow C++ API Reference | TensorFlow Core v2.7.0" ([https://www.tensorflow.org/api\\_docs/cc](https://www.tensorflow.org/api_docs/cc)). *TensorFlow*. Retrieved November 6, 2021.
46. "org.tensorflow | Java" ([https://www.tensorflow.org/api\\_docs/java/org/tensorflow/package-summary](https://www.tensorflow.org/api_docs/java/org/tensorflow/package-summary)). *TensorFlow*. Retrieved November 6, 2021.
47. Icaza, Miguel de (February 17, 2018). "TensorFlowSharp: TensorFlow API for .NET languages" (<https://github.com/migueldeicaza/TensorFlowSharp>). *GitHub*. Retrieved February 18, 2018.
48. Chen, Haiping (December 11, 2018). "TensorFlow.NET: .NET Standard bindings for TensorFlow" (<https://github.com/SciSharp/TensorFlow.NET>). *GitHub*. Retrieved December 11, 2018.
49. "haskell: Haskell bindings for TensorFlow" (<https://github.com/tensorflow/haskell>). tensorflow. February 17, 2018. Retrieved February 18, 2018.
50. Malmaud, Jon (August 12, 2019). "A Julia wrapper for TensorFlow" (<https://github.com/malmaud/TensorFlow.jl>). *GitHub*. Retrieved August 14, 2019. "operations like sin, \* (matrix multiplication), .\* (element-wise multiplication), etc [...]. Compare to Python, which requires learning specialized namespaced functions like tf.matmul."
51. "A MATLAB wrapper for TensorFlow Core" (<https://github.com/asteinh/tensorflow.m>). *GitHub*. November 3, 2019. Retrieved February 13, 2020.

52. "Use TensorFlow from Pascal (FreePascal, Lazarus, etc.)" (<https://github.com/zsoltszakaly/tensorflowforpascal>). *GitHub*. January 19, 2023. Retrieved January 20, 2023.
53. "tensorflow: TensorFlow for R" (<https://github.com/rstudio/tensorflow>). RStudio. February 17, 2018. Retrieved February 18, 2018.
54. Platanios, Anthony (February 17, 2018). "tensorflow\_scala: TensorFlow API for the Scala Programming Language" ([https://github.com/eaplatanios/tensorflow\\_scala](https://github.com/eaplatanios/tensorflow_scala)). *GitHub*. Retrieved February 18, 2018.
55. "rust: Rust language bindings for TensorFlow" (<https://github.com/tensorflow/rust>). tensorflow. February 17, 2018. Retrieved February 18, 2018.
56. Mazare, Laurent (February 16, 2018). "tensorflow-ocaml: OCaml bindings for TensorFlow" (<https://github.com/LaurentMazare/tensorflow-ocaml>). *GitHub*. Retrieved February 18, 2018.
57. "fazibear/tensorflow.cr" (<https://github.com/fazibear/tensorflow.cr>). *GitHub*. Retrieved October 10, 2018.
58. "tensorflow package - github.com/tensorflow/tensorflow/tensorflow/go - pkg.go.dev" (<https://pkg.go.dev/github.com/tensorflow/tensorflow/tensorflow/go>). *pkg.go.dev*. Retrieved November 6, 2021.
59. "Swift for TensorFlow (In Archive Mode)" (<https://www.tensorflow.org/swift/guide/overview>). *TensorFlow*. Retrieved November 6, 2021.
60. "TensorFlow.js | Machine Learning for JavaScript Developers" (<https://www.tensorflow.org/js>). *TensorFlow*. Retrieved October 28, 2021.
61. "TensorFlow Lite | ML for Mobile and Edge Devices" (<https://www.tensorflow.org/lite>). *TensorFlow*. Retrieved November 1, 2021.
62. "TensorFlow Lite" (<https://www.tensorflow.org/lite/guide>). *TensorFlow*. Retrieved November 1, 2021.
63. "TensorFlow Extended (TFX) | ML Production Pipelines" (<https://www.tensorflow.org/tfx>). *TensorFlow*. Retrieved November 2, 2021.
64. "Customization basics: tensors and operations | TensorFlow Core" (<https://www.tensorflow.org/tutorials/customization/basics>). *TensorFlow*. Retrieved November 6, 2021.
65. "Guide | TensorFlow Core" (<https://www.tensorflow.org/guide>). *TensorFlow*. Retrieved November 4, 2021.
66. "Libraries & extensions" (<https://www.tensorflow.org/resources/libraries-extensions>). *TensorFlow*. Retrieved November 4, 2021.
67. "Colaboratory – Google" (<https://research.google.com/colaboratory/faq.html>). *research.google.com*. Retrieved November 10, 2018.
68. "Google Colaboratory" (<https://colab.research.google.com/>). *colab.research.google.com*. Retrieved November 6, 2021.
69. Bradbury, James; Frostig, Roy; Hawkins, Peter; Johnson, Matthew James; Leary, Chris; MacLaurin, Dougal; Necula, George; Paszke, Adam; Vanderplas, Jake; Wanderman-Milne, Skye; Zhang, Qiao (June 18, 2022), "JAX: Autograd and XLA" (<https://web.archive.org/web/20220618205214/https://github.com/google/jax>), *Astrophysics Source Code Library*, Google, Bibcode:2021ascl.soft11002B (<https://ui.adsabs.harvard.edu/abs/2021ascl.soft11002B>), archived from the original (<https://github.com/google/jax>) on June 18, 2022, retrieved June 18, 2022
70. "Using JAX to accelerate our research" (<https://www.deepmind.com/blog/using-jax-to-accelerate-our-research>). *www.deepmind.com*. Archived (<https://web.archive.org/web/20220618205746/https://www.deepmind.com/blog/using-jax-to-accelerate-our-research>) from the original on June 18, 2022. Retrieved June 18, 2022.

71. "Why is Google's JAX so popular?" (<https://analyticsindiamag.com/why-is-googles-jax-so-popular/>). *Analytics India Magazine*. April 25, 2022. Archived (<https://web.archive.org/web/20220618210503/https://analyticsindiamag.com/why-is-googles-jax-so-popular/>) from the original on June 18, 2022. Retrieved June 18, 2022.
72. "Intelligent Scanning Using Deep Learning for MRI" (<https://blog.tensorflow.org/2019/03/intelligent-scanning-using-deep-learning.html>). Retrieved November 4, 2021.
73. "Case Studies and Mentions" (<https://www.tensorflow.org/about/case-studies>). *TensorFlow*. Retrieved November 4, 2021.
74. "Ranking Tweets with TensorFlow" (<https://blog.tensorflow.org/2019/03/ranking-tweets-with-tensorflow.html>). Retrieved November 4, 2021.
75. 3.5kshares; 72kreads. "A Complete Guide to the Google RankBrain Algorithm" (<https://www.searchenginejournal.com/google-algorithm-history/rankbrain/>). *Search Engine Journal*. Retrieved November 6, 2021.
76. "InSpace: A new video conferencing platform that uses TensorFlow.js for toxicity filters in chat" (<https://blog.tensorflow.org/2020/12/inspace-new-video-conferencing-platform-uses-tensorflowjs-for-toxicity-filters-in-chat.html>). Retrieved November 4, 2021.
77. Xulin. "流利说基于 TensorFlow 的自适应系统实践" ([http://mp.weixin.qq.com/s?\\_\\_biz=MzI0NjIzNDkwOA==&mid=2247484035&idx=1&sn=85fa0decac95e359435f68c50865ac0b&chksm=e94328f0de34a1e665e0d809b938efb34f0aa6034391891246fc223b7782ac3bfd6ddd588aa2#rd](http://mp.weixin.qq.com/s?__biz=MzI0NjIzNDkwOA==&mid=2247484035&idx=1&sn=85fa0decac95e359435f68c50865ac0b&chksm=e94328f0de34a1e665e0d809b938efb34f0aa6034391891246fc223b7782ac3bfd6ddd588aa2#rd)). *Weixin Official Accounts Platform*. Retrieved November 4, 2021.
78. "How Modiface utilized TensorFlow.js in production for AR makeup try on in the browser" (<https://blog.tensorflow.org/2020/02/how-modiface-utilized-tensorflowjs-in-ar-makeup-in-browser.html>). Retrieved November 4, 2021.
79. Byrne, Michael (November 11, 2015). "Google Offers Up Its Entire Machine Learning Library as Open-Source Software" (<https://www.vice.com/en/article/8q8avx/google-offers-up-its-entire-machine-learning-library-as-open-source>). *Vice*. Retrieved November 11, 2015.

---

Retrieved from "<https://en.wikipedia.org/w/index.php?title=TensorFlow&oldid=1137556769>"