

TensorFlow vs. PyTorch



TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

<https://www.tensorflow.org/>



TensorFlow vs. PyTorch

TensorFlow

Developed at Google by Google
Brain team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates static computation graph
on default mode

Must be defined before running

PyTorch

Developed at Facebook by
Facebook AI team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates dynamic computation
graph on default mode

Must be defined before running



TensorFlow vs. PyTorch

TensorFlow

Developed at Google by Google
Brain team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates static computation graph
on default mode

Must be defined before running

PyTorch

Developed at Facebook by
Facebook AI team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates dynamic computation
graph on default mode

Must be defined before running



TensorFlow vs. PyTorch

TensorFlow

Developed at Google by Google
Brain team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates static computation graph
on default mode

Must be defined before running

PyTorch

Developed at Facebook by
Facebook AI team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates dynamic computation
graph on default mode

Must be defined before running



TensorFlow vs. PyTorch

TensorFlow

Developed at Google by Google
Brain team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates static computation graph
on default mode

Must be defined before running

PyTorch

Developed at Facebook by
Facebook AI team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates dynamic computation
graph on default mode

Must be defined before running



TensorFlow vs. PyTorch

TensorFlow

Developed at Google by Google
Brain team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates static computation graph
on default mode

Must be defined before running

PyTorch

Developed at Facebook by
Facebook AI team

Tensors are the fundamental data
structures for computation

CUDA support for GPUs

Creates dynamic computation
graph on default mode

Must be defined before running



TensorFlow vs. PyTorch

TensorFlow

Network needs to be compiled

Code runs within `tf.Session` which separates it from Python

Debugging is done via `tfdg`

Visualization using built-in
TensorBoard

Deployment using TF Serving

PyTorch

There's no additional compilation step

Tightly integrated with Python

Native Python debugging

Visualization using `torchvision`,
`Matplotlib`, `seaborn`

Must be consumed via REST API



TensorFlow vs. PyTorch

TensorFlow

Network needs to be compiled

Code runs within `tf.Session` which separates it from Python

Debugging is done via `tfdg`

Visualization using built-in
TensorBoard

Deployment using TF Serving

PyTorch

There's no additional compilation step

Tightly integrated with Python

Native Python debugging

Visualization using `torchvision`,
`Matplotlib`, `seaborn`

Must be consumed via REST API



TensorFlow vs. PyTorch

TensorFlow

Network needs to be compiled

Code runs within `tf.Session` which separates it from Python

Debugging is done via `tfdg`

Visualization using built-in
TensorBoard

Deployment using TF Serving

PyTorch

There's no additional compilation step

Tightly integrated with Python

Native Python debugging

Visualization using `torchvision`,
`Matplotlib`, `seaborn`

Must be consumed via REST API



TensorFlow vs. PyTorch

TensorFlow

Network needs to be compiled

Code runs within `tf.Session` which separates it from Python

Debugging is done via `tfdg`

Visualization using built-in
TensorBoard

Deployment using TF Serving

PyTorch

There's no additional compilation step

Tightly integrated with Python

Native Python debugging

Visualization using `torchvision`,
`Matplotlib`, `seaborn`

Must be consumed via REST API



TensorFlow vs. PyTorch

TensorFlow

Network needs to be compiled

Code runs within `tf.Session` which separates it from Python

Debugging is done via `tfdg`

Visualization using built-in
TensorBoard

Deployment using TF Serving

PyTorch

There's no additional compilation step

Tightly integrated with Python

Native Python debugging

Visualization using `torchvision`,
`Matplotlib`, `seaborn`

Must be consumed via REST API



TensorFlow now has eager execution mode for dynamic computation graphs

PyTorch has also added support for static computation graphs

