1.0 INTRODUCTION TO PyTorch

24 August 2025 02:47 AM Avinash Yadav

2002 → Torch ————→ can be used to perform tensor based operations

↓

powerful scientific framework came in picture in 2002

↓

These tensor based operations could be performed on GPUs as well.

So researcher started using it to build deep learning applications as well. Then lots of neural network's implimention was done in Torch → AlexNet, VGGNet etc.

2 biggest limitations of Torch:

① Torch was 'Lua' based framework. That means whole whole Torch was written in Lua programming language.

⇒ So, if we wanted to build any of the application using Torch, then we had to code in Torch.

② The computational graph that were used in Torch were static in nature.

Then to resolve this issue, Meta AI researcher came up with a new library named 'PyTorch' that combines the capabilities of 'Torch' and the most common coding language among researchers i.e. 'Python'

PyTorch OVERVIEW

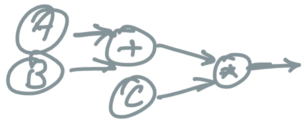
- **Open-Source Deep Learning Library:** Developed by Meta AI (formerly Facebook AI Research).
- **Python & Torch:** Combines Python's ease of use with the efficiency of the Torch scientific computing framework, originally built with Lua. Torch was known for high-performance tensor-based operations, especially on GPUs.

PyTorch RELEASE TIMELINE

PyTorch 0.1 (2017)

a visual way to represent mathematical operation

$(A+B) * C$



- *Key Features:*
  - Introduced the dynamic computation graph, enabling more flexible model architectures.
  - Seamless integration with other Python libraries (e.g., numpy, scipy).

- *Impact:*
  - Gained popularity among researchers due to its intuitive, Pythonic interface and flexibility.
  - Quickly featured in numerous research papers.

PyTorch 1.0 (2018)

- *Key Features:*
  - Bridged the gap between research and production environments.
  - Introduced TorchScript for model serialization and optimization.

    - Improved performance with Caffe2 integration.

- *Impact:*
    - Enabled smoother transitions of models from research to deployment.

PyTorch 1.x Series

- *Key Features:*
    - Support for distributed training.
    - ONNX(Open Neural Network Exchange) compatibility for interoperability with other frameworks.
    - Introduced quantization for model compression and efficiency.
    - Expanded ecosystem with torchvision (CV), torchtext (NLP), and torchaudio (audio).

- *Impact:*
    - Increased adoption by the research community and industry.
    - Inspired community libraries like PyTorch Lightning and Hugging Face Transformers.
    - Strengthened cloud support for easy deployment.

PyTorch 2.0

- *Key Features:*
    - Significant performance improvements in terms of latency and throughput.
    - Enhanced support for deployment and production-readiness.
    - Optimized for modern hardware (TPUs, custom AI chips).

- *Impact:*
    - Improved speed and scalability for real-world applications.
    - Better compatibility with a variety of deployment environments.