

# **PyTorch Resources Introduction**



## Joe Spisak

Joseph is the product lead for Facebook's open-source AI platform, including PyTorch and ONNX. His work spans internal collaborations with teams such as Oculus, Facebook Integrity, and FAIR, as well as working with the AI developer community to bring scalable tools to help push the state of the art forward. Prior to Facebook, Joseph led a deep-learning product management team and Global AI partnerships for Amazon AI. Before that, he was director of machine learning segment strategy for Intel.

### Lecturer Introduction

FACEBOOK AI

Georgia Tech

# Outline

- ◆ PyTorch Developer Resources
- ◆ Community Support
- ◆ Using Colab
- ◆ PyTorch Class resources

 PyTorch

# FROM RESEARCH TO PRODUCTION

An open source machine learning framework that accelerates the path from research prototyping to production deployment.

[Get Started >](#)
[Click Here to Read About Latest Updates and Improvements to PyTorch's Tutorials](#)

## KEY FEATURES &

[See all Features >](#)
**MiDaS**

The MiDaS v2 model for computing relative depth from a single image.


**ntsn**

classify birds using this fine-grained image classifier


**DeepLabv3-ResNet101**

DeepLabV3 model with a ResNet-101 backbone


**Transformer (NMT)**

Transformer models for English-French and English-German translation.


**WaveGlow**

WaveGlow model for generating speech from mel spectrograms (generated by Tacotron2)


**ResNext WSL**

ResNext models trained with billion scale weakly-supervised data.


**AdverTorch**

A toolbox for adversarial robustness research. It contains modules for generating adversarial examples and defending against attacks.

**Allegro Trains**

Allegro Trains is a full system ML / DL experiment manager, versioning and ML-Ops solution.

**baal**

baal (bayesian active learning) aims to implement active learning using metrics of uncertainty derived from approximations of bayesian posteriors in neural networks.

**Captum**
**BoTorch**
**Albummentations**

Fast and extensible image augmentation library for different CV tasks like classification, segmentation, object detection and pose estimation.

**AllenNLP**

AllenNLP is an open-source research library built on PyTorch for designing and evaluating deep learning models for NLP.

**Catalyst**
**Shortcuts**

## Prerequisites

macOS Version

## Python

## Package Manager

## Installation

Anaconda

pip

## Verification

## Building from source

## Prerequisites

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, 1.5 builds that are generated nightly. Please ensure that you have met the prerequisites below (e.g., numpy), depending on your package manager. Anaconda is our recommended package manager since it installs all dependencies. You can also [install previous versions of PyTorch](#). Note that LibTorch is only available for C++.

PyTorch Build	Stable (1.5.1)	Preview (Nightly)	
Your OS	Linux	Mac	Windows
Package	Conda	Pip	LibTorch
Language	Python		C++/Java
CUDA	9.2	10.1	10.2
Run this Command:			
<code>curl https://pytorch.org/get-pytorch.py   python</code>			
<small>* Mac OS Binaries don't support CUDA, install from source if CUDA is needed</small>			

# Active community with >31k users and over 206k posts

The screenshot shows the PyTorch discussion forum interface. At the top, there is a navigation bar with links for "all categories", "Latest" (which is highlighted in red), "New (88)", "Unread (1)", "Top", "Categories", and a "New Topic" button. Below the navigation bar is a search bar and a user profile icon.

The main content area displays a list of topics. Each topic entry includes the title, a small preview of the first post, the number of replies, views, and the time since the last reply. The topics listed are:

- Quantization.convert after QAT pickling issue • 2 replies, 21 views, 1m ago. (Tags: quantization)
- Duplicate fully connected layers and train model with new duplicated layers only • 2 replies, 22 views, 21m ago. (Tags: vision)
- classification problem in pytorch with loss function CrossEntropyLoss returns negative output in prediction • 5 replies, 136 views, 29m ago.
- Net.cuda() move all parameters to GPU, then how to ensure the autograd? • 2 replies, 21 views, 33m ago.
- How to use “break” in DistributedDataParallel training • 4 replies, 28 views, 35m ago. (Tags: distributed-rpc)
- Is transforms.RandomResizedCrop used for Data Augmentation? • 3 replies, 9.7k views, 38m ago. (Tags: vision)
- Routing by agreement for NMT • 0 replies, 5 views, 41m ago. (Tags: nlp)
- How to combine multiple criterions to a loss function? • 36 replies, 41.4k views, 44m ago.

# PyTorch has native support for Google Colab – Free, Supports GPUs & Jupyter

The screenshot shows the Google Colab interface. At the top, there's a navigation bar with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help' options. On the right of the bar are 'Share', 'Settings', and a user profile icon. Below the bar is a toolbar with '+ Code', '+ Text', and 'Copy to Drive' buttons. To the right of the toolbar are 'Connect', 'Editing' (with a pencil icon), and a collapse/expand button.

**Table of contents** (left sidebar):

- Getting started
- Data science
- Machine learning
- More Resources
- Machine Learning Examples
- Section

**What is Colaboratory?** (main content area):

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

**Getting started** (section header):

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

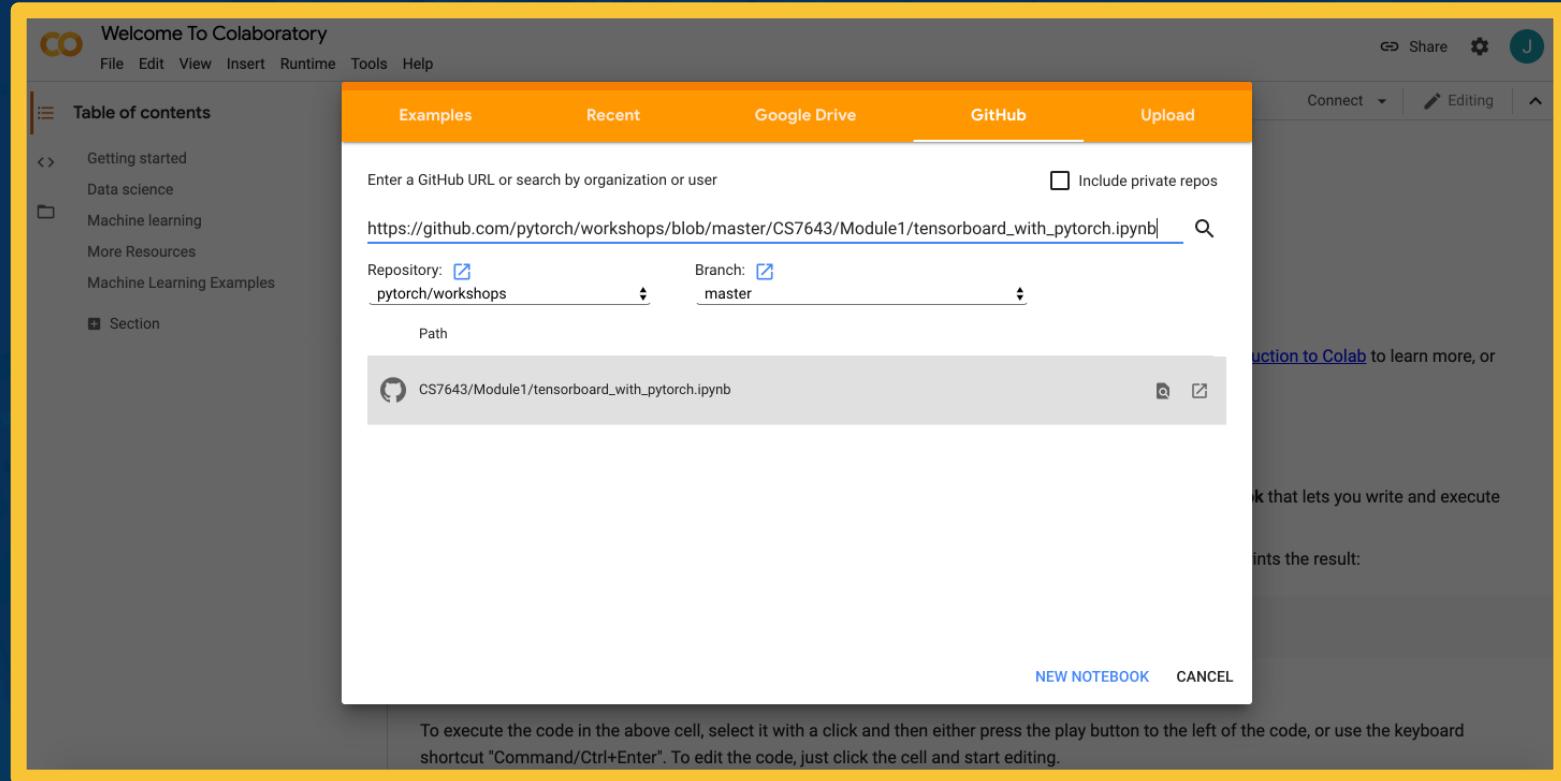
To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Using Colab | <https://colab.research.google.com/>

FACEBOOK AI

Georgia Tech

# Option 1: Loading a notebook from GitHub

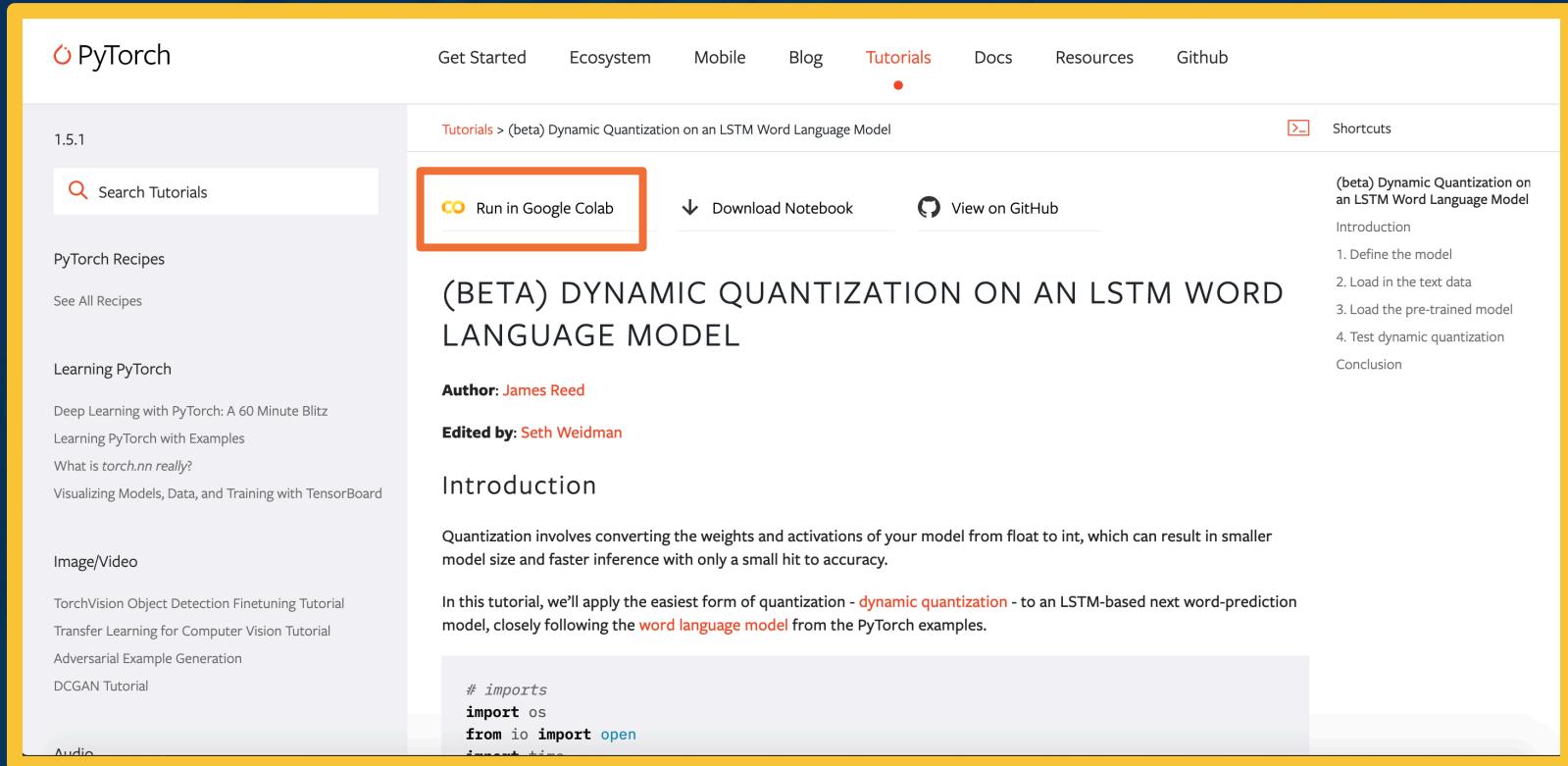


Using Colab | Enter the GitHub repo

FACEBOOK AI

Georgia Tech

## Option 2: Click through on PyTorch.org



The screenshot shows the PyTorch.org website with a yellow header bar. The navigation menu includes 'Get Started', 'Ecosystem', 'Mobile', 'Blog', 'Tutorials' (which is the active page), 'Docs', 'Resources', and 'Github'. Below the header, there's a sidebar with sections like 'PyTorch Recipes' (including 'Search Tutorials'), 'Learning PyTorch' (with links to 'Deep Learning with PyTorch: A 60 Minute Blitz', 'Learning PyTorch with Examples', 'What is torch.nn really?', and 'Visualizing Models, Data, and Training with TensorBoard'), 'Image/Video' (with links to 'TorchVision Object Detection Finetuning Tutorial', 'Transfer Learning for Computer Vision Tutorial', 'Adversarial Example Generation', and 'DCGAN Tutorial'), and 'Audio'. The main content area shows the title '(BETA) DYNAMIC QUANTIZATION ON AN LSTM WORD LANGUAGE MODEL' by James Reed, edited by Seth Weidman. It features an 'Introduction' section with text about quantization and a code snippet in a light gray box:

```
# imports
import os
from io import open
import time
```

Using Colab | Enter the GitHub repo

FACEBOOK AI

Georgia Tech

# PyTorch on Jupyter via Colab

The screenshot shows a Google Colab notebook interface. The title bar reads 'tensorboard\_with\_pytorch.ipynb'. The menu bar includes File, Edit, View, Insert, Runtime, Tools, and Help. The toolbar has options for Code, Text, Copy to Drive, Share, and a user icon. A code cell at the top contains the command '%matplotlib inline'. Below it, a section titled 'How to use TensorBoard with PyTorch' is expanded, containing text about TensorBoard's features and its use with PyTorch. A subsection 'Installation' is shown with instructions for installing PyTorch via Anaconda or pip. Another section 'Using TensorBoard in PyTorch' is partially visible at the bottom. A note at the bottom of the page says 'Let's now try using TensorBoard with PyTorch! Before logging anything, we need to create a `SummaryWriter` instance.'

tensorboard\_with\_pytorch.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive Connect Editing

%matplotlib inline

How to use TensorBoard with PyTorch

TensorBoard is a visualization toolkit for machine learning experimentation. TensorBoard allows tracking and visualizing metrics such as loss and accuracy, visualizing the model graph, viewing histograms, displaying images and much more. In this tutorial we are going to cover TensorBoard installation, basic usage with PyTorch, and how to visualize data you logged in TensorBoard UI.

### Installation

PyTorch should be installed to log models and metrics into TensorBoard log directory. The following command will install PyTorch 1.4+ via Anaconda (recommended):

```
:::  
$ conda install pytorch torchvision -c pytorch  
or pip  
:::  
$ pip install torch torchvision
```

### Using TensorBoard in PyTorch

Let's now try using TensorBoard with PyTorch! Before logging anything, we need to create a `SummaryWriter` instance.

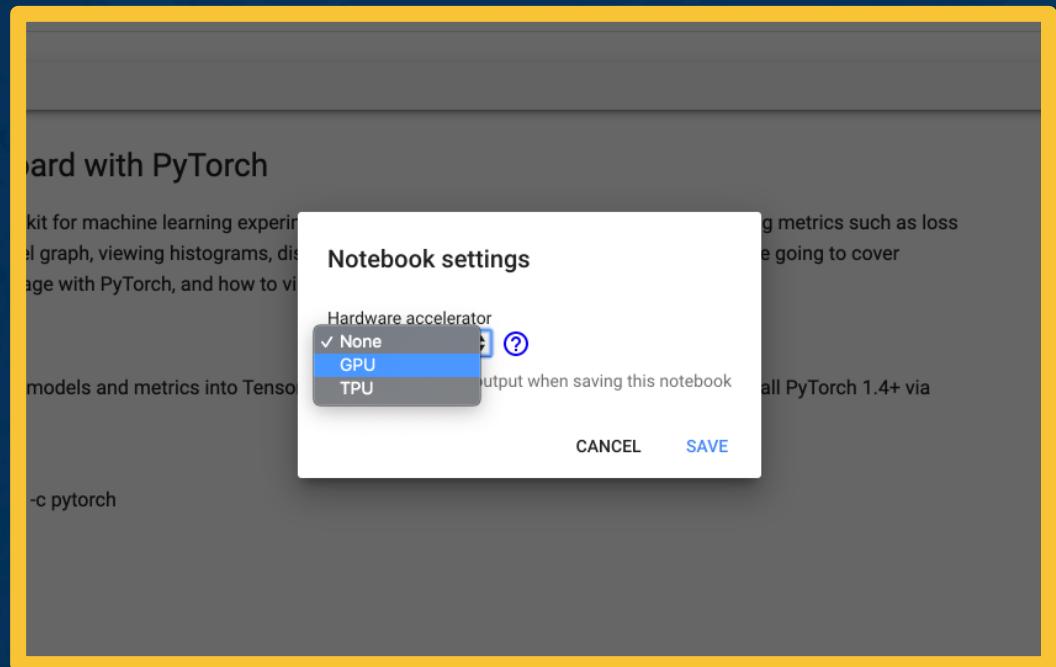
Using Colab | Notebook is now live!

FACEBOOK AI

Georgia Tech

Change your runtime to GPU or TPU... ☺, Shift-Enter and you're off!!

- ◆ CPU (Default)
- ◆ GPU – typically a P100 (9.3 Tflops)
- ◆ TPU (Tensor Processing Units) – 90TOPS (v3)

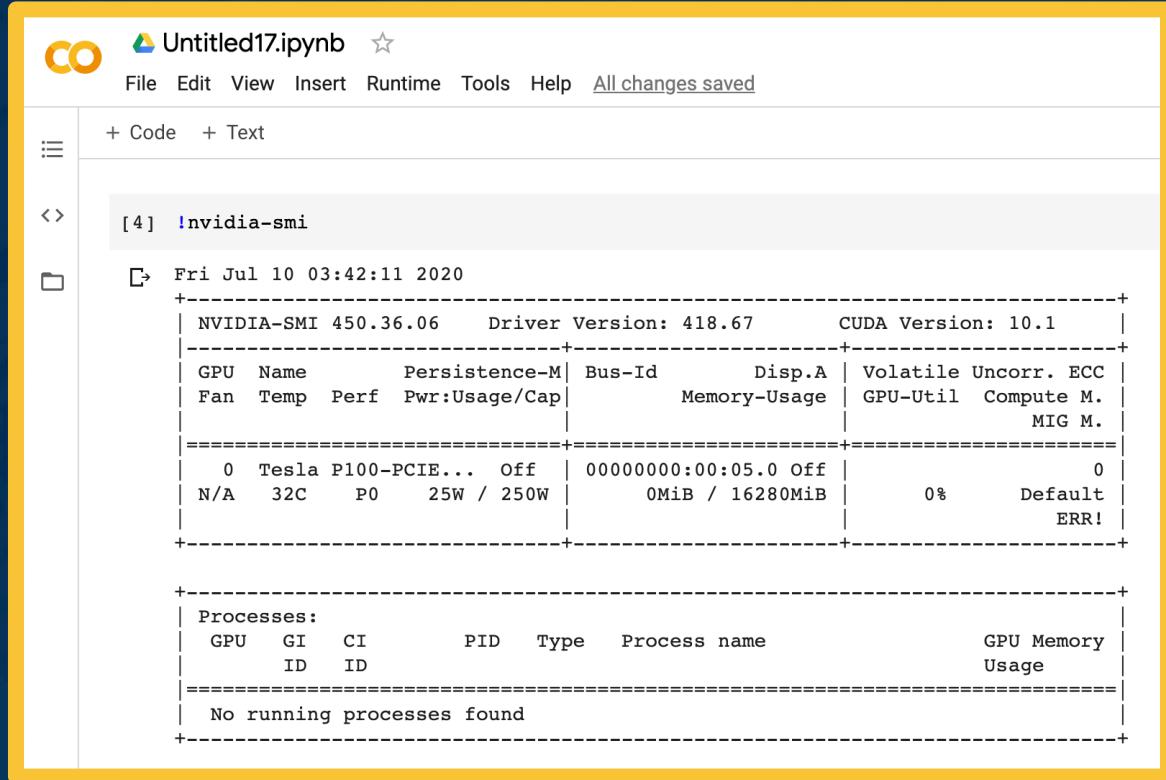


Using Colab | Select runtime (GPU is preferred)

FACEBOOK AI

Georgia Tech

# Change your runtime to GPU and run ‘!nvidia-smi’



The screenshot shows a Google Colab notebook titled "Untitled17.ipynb". The runtime is set to GPU. Cell [4] contains the command "!nvidia-smi". The output is as follows:

```
[4]: !nvidia-smi
Fri Jul 10 03:42:11 2020
+-----+
| NVIDIA-SMI 450.36.06    Driver Version: 418.67      CUDA Version: 10.1 |
+-----+
| GPU  Name        Persistence-M | Bus-Id     Disp.A  | Volatile Uncorr. ECC | | | | |
| Fan  Temp     Perf  Pwr:Usage/Cap | Memory-Usage | GPU-Util Compute M. |
|          |          |          |          |          |          | MIG M. |
+-----+
| 0  Tesla P100-PCIE... Off | 00000000:00:05.0 Off |           0 |
| N/A   32C    P0    25W / 250W |          0MiB / 16280MiB |      0%  Default |
|                               |                           |          ERR! |
+-----+
+-----+
| Processes:
| GPU  GI  CI          PID   Type  Process name          GPU Memory |
|          ID  ID          ID          ID                 Usage  |
+-----+
| No running processes found
+-----+
```

Pointers available at: <https://github.com/pytorch/workshops/tree/master/CS7643>

pytorch / workshops

Unwatch 5 Unstar 5 Fork 5

Code Issues Pull requests 3 Actions Projects Wiki Security Insights Settings

Branch: master workshops / CS7643 / Go to file Add file ▾

jspisak committed 1b9fe7c 3 minutes ago History

..

Module1	Update readme.md	3 minutes ago
Module2	Create readme.md	14 hours ago
Module3	Create readme.md	14 hours ago
Module4	Create readme.md	14 hours ago
readme.md	Update readme.md	14 hours ago

# Thanks!