Assignment 1 - All About torch.Tensor

Deep Learning with PyTorch: Zero to GANs

"A journey of a thousand miles begins with a single step"

Congrats on making it through the first lecture! Before we jump head-first into deep learning, it is worth spending some to familiarize ourselves with the tools, libraries and concepts we are going to build upon during the course. The key objective of this assignment is to develop a solid understanding of PyTorch tensors.

Learning Goals

You will learn the following by doing this assignment:

1. Writing code in Python
2. Working with Jupyter notebooks
3. Working with tensors in PyTorch
4. Reading the documentation of an open-source library
5. Versioning and sharing Jupyter notebooks with Jovian.ml
6. Writing a blog post (the first post ever, for many of you)
7. Sharing your work online & getting feedback
8. Learning from others' work & giving constructive feedback

Problem Statement

In this assignment, you will pick 5 interesting functions related to PyTorch tensors by reading the documentation, create a Jupyter notebook using a starter template to illustrate their usage, write a blog post to provide some explanation about your code, and share it with the community.

Follow the step-by-step explanations given below in the **Detailed Guidelines** section, and you should sail through the assignment! If you get stuck at any point, look through the **Reference Links** section, or ask a question on the official assignment thread on the forum.

Submission Instructions

You need to make your submission via this Google form:
https://forms.gle/8QvFH64gU99VkivA8

Provide the links to your **Jupyter notebook** and **blog post** in the form. In case you were unable to create a blog post, you can add more detailed explanation within your Jupyter notebook itself, and provide the same link in the "Blog post URL" text box. However, we highly recommend writing a separate blog post.

For your submission to be accepted, your Jupyter notebook must contain some explanation and examples for at least 5 PyTorch functions related to **torch.Tensor**.

Reference Links

Official links for this assignment:

- Forum thread: https://jovian.ml/forum/t/assignment-1-all-about-torch-tensor/1559 (ask questions and get help here)
- Starter notebook: https://jovian.ml/aakashns/01-tensor-operations

- Submission form: https://forms.gle/8QvFH64gU99VkivA8
- Lecture 1 forum thread: https://jovian.ml/forum/t/lecture-1-pytorch-basics-linear-regression/1541
- Share and embed Jupyter notebooks online with Jovian: https://medium.com/@aakashns/share-and-embed-jupyter-notebooks-online-with-jovian-ml-df709a03064e
- Jovian docs: https://jovian.ml/docs/user-guide/upload.html

Other useful resources:

- Tutorial on PyTorch tensors: https://www.analyticsvidhya.com/blog/2019/09/introduction-to-pytorch-from-scratch/
- Why you should blog: https://medium.com/@racheltho/why-you-yes-you-should-blog-7d2544ac1045
- Tips for writing a good blog post: https://www.freecodecamp.org/news/how-to-write-a-great-technical-blog-post-414c414b67f6/
- Python tutorial: https://www.w3schools.com/python/
- Jupyter tutorial: https://jovian.ml/aakashns/jupyter-tutorial

Detailed Guidelines

Step 1: Browse through PyTorch documentation and pick 5 interesting functions

- Visit the official documentation page for torch.tensor: https://pytorch.org/docs/stable/tensors.html
- Go through the page and identify 5 interesting functions. Don't pick the first 5, and try to pick functions that are related to each other (e.g. matrix operations, derivatives/gradients, tensor creation, gradient computation)
- Read the documentation and examples to understand their usage. You can also try out the functions online by executing the notebook from Lesson 1 and modifying the code there.

Step 2: Create a Jupyter notebook with explanation & examples for the selected functions

- Open up the starter notebook and click the "Fork" button to copy it to your profile: https://jovian.ml/aakashns/01-tensor-operations
- Change the visibility of your notebook from "Private" to "Public" by clicking on the "Public" badge below the project title.
- Execute your notebook by clicking the "Run" button and selecting "Run on Binder". It will take Binder a few minutes to install all the required libraries and start the Jupyter notebook.
- Open up the Jupyter notebook, and add some explanation and examples for each of the selected functions. You can use the structure given in the notebook, or create your own. Try to get creative!
- Don't copy-paste examples from the docs. Come up with your own. Also, add at least one example per function which leads to an error, and explain how things can go wrong while using the function.
- You don't have to give detailed explanations in your notebook. A 2-line explanation will do. You can provide detailed explanation in your blog post.

Note: Binder is a free service, which allows you to run Jupyter notebooks online for a limited amount of time. If you leave your notebook idle for a long time, it may shut down. Read the Binder FAQ to learn more. We recommend that you commit your notebook to Jovian regularly by running the last 3 cells (see Step 3)

Step 3: Publish your Jupyter notebook online to your Jovian.ml profile

- Executing the last 3 cells of the notebook will publish your Jupyter notebook online to Jovian. Every time you run jovian.commit(), a new version will be recorded. Commit regularly to make sure you don't lose your work

- Once you have completed the notebook, commit the final version to Jovian, and move on to the next step. If you ever need to update your notebook, just click “Run on Binder” again.

Step 4: Write a blog post to accompany your notebook

- Sign up on medium.com or any other free blogging platform of your choice. We recommend medium because it is really easy to use and has millions of readers who can find your blog.
- Create a blog post to accompany your notebook. It will have the same structure as the notebook, but you can add more detailed explanation and pictures to make it easier to read and understand, especially for beginners.
- Within your blog post, embed cells from your Jupyter notebook hosted on Jovian.ml to allow readers to view the code & outputs a they would appear within Jupyter. If you are unable to embed, you can add screenshots instead, and link to your Jupyter notebook at the beginning or end of the blog post

Read the “Blogging Guidelines” section below for more help on creating a blog post.

Step 5: Share your work with the community

- Showcase your work to the the community by posting it to the “Share Your Work” topic on the course forum. Post a link to your Jupyter notebook or blog post, and provide some explanation about what you’ve done. Others will get a chance to
- (Optional) Share your work on social media. You can do this using the “Share” button on your Jupyter notebook.
  - **Twitter**: Tag @JovianML while tweeting your notebook/blog. We’ll retweet 5 great notebooks every day. Use the hashtag #zerotogans
  - **LinkedIn**: Share with your work with your professional network and get their feedback
  - **Facebook**: Share your work with your friends and colleagues

Step 6: Read posts from other course participants & give feedback

- Visit the “Share Your Work” topic on the course forum, and check out what other course participants have shared.
- Try to read at least 5 other blog posts and give constructive feedback.
- Show some appreciation by linking the forum post, pressing the Star button on the notebook page, or giving a clap on Medium. Maybe even leave a helpful comment!

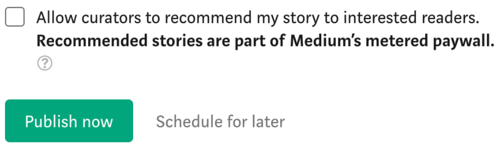
Step 7: Submit your work using the submission form

Don’t forget to make a submission using this Google form: https://forms.gle/8QvFH64gU99VkivA8

Blogging Guidelines

Read Why you should blog and tips for write a good blog post to understand why blogging is an important part of learning and how you it will help you build your professional profile.

A couple of notes about Medium:
1. To create a blog post on Medium, you need to select the “New Story” option from the profile menu on the top right.
2. While publishing your blog, **remove the tick mark** from the “Allow curators to recommend my story..” checkbox. If you tick this, others may not be able to read your blog



□ Allow curators to recommend my story to interested readers.
Recommended stories are part of Medium’s metered paywall.

You can use this structure as a reference for creating your blog post:

- Interesting Title & Subtitle
- Introduction to PyTorch
- Section 1
    - Explanation
    - Examples (with embedded code cells)
- Section 2
    - Explanation
    - Examples
- Section 3
    - Explanation
    - Examples
- Section 4
    - Explanation
    - Examples
- Section 5
    - Explanation
    - Examples
- Conclusion
- Links for Reference & Further Reading
- Links to your social profiles

You don’t have to stick to this structure, use your imagination and creativity to make it interesting.

Come up with an interesting title for your blog post. Here are some examples:

- “All about matrix operations in PyTorch”
- “5 PyTorch functions you didn’t know you needed”
- “A beginner’s guide to tensor broadcasting in PyTorch”
- “Interesting ways to create tensors in PyTorch”
- “Data types in PyTorch and how to switch between them”
- “How to calculate gradients automatically in PyTorch”
- “Tensor arithmetic in PyTorch explained with code”