Task 1 – Extra Code Instructions

The objective of this document is to specify and clarify some important instructions when writing code.

Important Instuctions:

1. One of the tasks have a **restriction on usage of libraries.** You can use only **PyTorch Tensor API** in **Task 1A**. There is no restriction for **Task 1B** other than that you cannot use libraries apart from the ones we have imported in the Python scripts. If you are doubtful about use of some library, you can ask us on Piazza.

2. You should **not change names of already specified functions and modules.** You can add more functions and modules if you want to.

3. You have to **strictly follow instructions inside the code**. Stick to the variable names if you are specifically instructed to use. You have to exactly return said attributes from the function and use defined inputs. **Read docstrings (or code comments) for instructions**.

4. **Use unit tests to find out if your code matched our criteria.** We have given tests for some tasks that strictly need to follow this rule. We will roll more tests for verification if required. **Announcement regarding this will be made on Piazza.**

5. You should **change or add to parts of the code which are and may look incomplete.** For example, function may look complete but in theory is not completely doing what is expected of it. Use your conscience to make a judgement if your code is complete or not. You can use test module to write more test cases for each function. We have also mentioned some explicit **TODOs** where it may not be obvious that you have to make a change.

6. You should **not change parts of code that we have written**. Most of the code that we have left incomplete is very minimal like variable names or hints for you and should try to complete it. **Sometimes it may not be of an issue** if you change code written by us. If you have a reason to change the code and cannot live without doing it, post is as a query on Piazza. **At some places, we have marked if something should not be changed**.

Coding's best practices for extra marks:

1. Code should be **readable and easily understandable.**
2. Document your code with **docstrings** like we have done.
3. Use **OOP concepts and modularize code** if you can reason the usage properly.
4. **Vectorize operations** to make your **code run faster**. A significant improvement in speed will sure fetch more marks.
5. **Writing rigorous unit tests is rewarding**.
6. Use a **text-editor like VS Code**. You can install some cool extensions like TODO finder, so that it makes your work easy.

Resources:

1. **Docstring:** Follow provided examples in code
2. **Unit testing in Python:** YouTube link

…Best Wishes ! …