# Contributing to Torch7 Core (torch7, nn, cutorch, cunn)

Thanks a lot! There are plenty of ways you can help!

Please take a moment to review this document in order to make the contribution

process easy and effective for everyone involved.

Following these guidelines helps to communicate that you respect the time of

the developers managing and developing this open source project. In return,

they should reciprocate that respect in addressing your issue or assessing

patches and features.

## Using the issue tracker

The [issue tracker](https://github.com/torch/torch7/issues) is

the preferred channel for [bug reports](#bugs), [features requests](#features)

and [submitting pull requests](#pull-requests), but please respect the following

restrictions:

\* Please \*\*do not\*\* use the issue tracker for personal support requests (use

[mailing-list](https://groups.google.com/forum/#!forum/torch7)).

\* Please \*\*do not\*\* open issues regarding the code in a torch package

outside the core. For example don't open issues about the

REPL in the torch7 issue tracker, use the trepl issue tracker for that.

<a name="bugs"></a>

## Bug reports

A bug is a \_demonstrable problem\_ that is caused by the code in the repository.

Good bug reports are extremely helpful - thank you!

Guidelines for bug reports:

1. \*\*Use the GitHub issue search\*\* &mdash; check if the issue has already been

reported.

2. \*\*Check if the issue has been fixed\*\* &mdash; try to reproduce it using the

latest `master` or development branch in the repository.

3. \*\*Isolate the problem\*\* &mdash; ideally create test case that is within reason,

preferably within 100 lines of code.

A good bug report shouldn't leave others needing to chase you up for more

information. Please try to be as detailed as possible in your report. What is

your environment? What steps will reproduce the issue? What OS do you

experience the problem? What would you expect to be the outcome? All these

details will help people to fix any potential bugs.

<a name="features"></a>

## Feature requests

Feature requests are welcome to be filed. Torch is community-developed,

the maintainers are not exclusive torch developers, so keep that in mind.

The purpose of feature requests is for others who are looking to implement

a feature are aware of the interest in the feature.

<a name="pull-requests"></a>

## Pull requests

Good pull requests - patches, improvements, new features - are a fantastic

help. They should remain focused in scope \*\*and avoid containing unrelated

commits.\*\*

\*\*Please ask first\*\* before embarking on any significant pull request (e.g.

implementing features, refactoring code, porting to a different language),

otherwise you risk spending a lot of time working on something that the

project's developers might not want to merge into the project.

Please adhere to the coding conventions used throughout a project (indentation,

accurate comments, etc.) and any other requirements (such as test coverage).

Adhering to the following this process is the best way to get your work

included in the project:

1. [Fork](https://help.github.com/articles/fork-a-repo) the project, clone your

fork, and configure the remotes:

```bash

# Clone your fork of the repo into the current directory

git clone https://github.com/<your-username>/torch7.git

# Navigate to the newly cloned directory

cd torch7

# Assign the original repo to a remote called "upstream"

git remote add upstream https://github.com/torch/torch7.git

```

2. If you cloned a while ago, get the latest changes from upstream:

```bash

git checkout master

git pull upstream master

```

3. Create a new topic branch (off the main project development branch) to

contain your feature, change, or fix:

```bash

git checkout -b <topic-branch-name>

```

4. Commit your changes in logical chunks. Please try to adhere to these [git commit

message guidelines](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html)

. Use Git's [interactive rebase](https://help.github.com/articles/about-git-rebase)

feature to tidy up your commits before making them public. This helps us keep the

commit history in logical blocks and clean, as torch grows.

For example:

- If you are adding a new function or a module, keep the module + tests + doc

to a single commit unless logically warranted.

- If you are fixing a bug, keep the bugfix to a single commit unless logically warranted.

5. Locally merge (or rebase) the upstream development branch into your topic branch:

```bash

git pull [--rebase] upstream master

```

6. Push your topic branch up to your fork:

```bash

git push origin <topic-branch-name>

```

7. [Open a Pull Request](https://help.github.com/articles/using-pull-requests/)

with a clear title and description.

\*\*IMPORTANT\*\*: By submitting a patch, you agree to allow the project owners to

license your work under the terms of the BSD License.