
octopod Documentation

Release 3.2.0

ShopRunner data science team

Apr 18, 2022

CONTENTS

1	Structure	3
2	Installation	5
3	Notes	7
4	Development	9
4.1	Core Octopod	9
4.2	Octopod Ensemble	10
4.3	Learner Utils	10
4.4	Octopod Text	10
4.5	Octopod Vision	10
4.6	Contributing and Making PRs	10
4.7	Contributor Covenant Code of Conduct	12
	Python Module Index	15
	Index	17



Octopod is a general purpose deep learning library developed by the ShopRunner Data Science team to train multi-task image, text, or ensemble (image + text) models.

What differentiates our library is that you can train a multi-task model with different datasets for each of your tasks. For example, you could train one model to label dress length for dresses and pants length for pants.

See the [docs](#) for more details.

To quickly get started, check out one of our tutorials in the `notebooks` folder. In particular, the `synthetic_data` tutorial provides a very quick example of how the code works.

Note 7/08/20: We are renaming this repository Octopod (previously called Tonks). The last version of the PyPI library under the name Tonks will not break but will warn the user to begin installing and using Octopod instead. No further development will continue under the name Tonks.

Note 6/12/20: Our team previously had a tradition of naming projects with terms or characters from the Harry Potter series, but we are disappointed by J.K. Rowling's persistent transphobic comments. In response, we will be renaming this repository, and are working to develop an inclusive solution that minimizes disruption to our users.

STRUCTURE

- notebooks
 - `fashion_data`: a set of notebooks demonstrating training Octopod models on an open source fashion dataset consisting of images and text descriptions
 - `synthetic_data`: a set of notebooks demonstrating training Octopod models on a set of generated color swatches. This is meant to be an easy fast demo of the library's capabilities that can be run on CPU's.
- octopod
 - `ensemble`: code for ensemble models of text and vision models
 - `text`: code for text models with a BERT architecture
 - `vision`: code for vision models with ResNet50 architectures

INSTALLATION

```
pip install octopod
```

You may get an error from the `tokenizer` package if you do not have a Rust compiler installed; see <https://github.com/huggingface/transformers/issues/2831#issuecomment-592724471>.

NOTES

Currently, this library supports ResNet50 and BERT models.

In some of our documentation the terms `pretrained` and `vanilla` appear. `pretrained` is our shorthand for Octopod models that have been trained at least once already so their weights have been tuned for a specific use case. `vanilla` is our shorthand for base weights coming from `transformers` or `PyTorch` for the out-of-the-box BERT and ResNet50 models.

For our examples using text models, we use the `transformers` repository managed by huggingface. The most recent version is called `transformers`. The huggingface repo is the appropriate place to check on BERT documentation and procedures.

DEVELOPMENT

Want to add to or fix issues in Octopod? We welcome outside input and have tried to make it easier to test. You can run everything inside a docker container with the following:

```
# to build the container
# NOTE: this may take a while
docker build -t octopod .
# nvidia-docker run : basic startup with nvidia docker to access gpu
# --rm : deletes container when closed
# -p : exposes ports (ex: for jupyter notebook to work)
# bash : opens bash in the container once it starts
# "pip install jupyter && bash" : install requirements-dev and bash
nvidia-docker run \
  -it \
  --rm \
  -v "${PWD}:/octopod" \
  -p 8888:8888 \
  octopod /bin/bash -c "pip install jupyter && bash"
# run jupyter notebook
jupyter notebook --ip 0.0.0.0 --no-browser --allow-root --NotebookApp.token='' --
↳NotebookApp.password=''
```

4.1 Core Octopod

Core Octopod is made up of a learner and dataloader class designed for multi-task multi-dataset learning.

4.1.1 Multitask Learner

4.1.2 Multitask Dataloader

```
class octopod.dataloader.MultiDatasetLoader (loader_dict, shuffle=True)
    Load datasets for multiple tasks
```

Parameters

- **loader_dict** (*dict*) – dictionary of DataLoaders
- **shuffle** (*Boolean (defaults to True)*) – Flag for whether or not to shuffle the data

4.2 Octopod Ensemble

The ensemble aspects of Octopod are housed here. This includes sample model architectures, dataset class, and helper functions.

4.2.1 Model Architectures

4.2.2 Dataset

4.3 Learner Utils

This section contains helper code for the Octopod learner pipelines for supporting multiple loss functions and metrics for individual tasks.

4.3.1 Metric Utils

4.4 Octopod Text

The text aspects of Octopod are housed here. This includes sample model architectures and a dataset class.

4.4.1 Model Architectures

4.4.2 Dataset

4.5 Octopod Vision

The computer vision aspects of Octopod are housed here. This includes sample model architectures, dataset class, and helper functions.

4.5.1 Model Architectures

4.5.2 Dataset

4.5.3 Helper Functions

4.6 Contributing and Making PRs

4.6.1 How to Contribute

We welcome contributions in the form of issues or pull requests!

We want this to be a place where all are welcome to discuss and contribute, so please note that this project is released with a Contributor Code of Conduct. By participating in this project you agree to abide by its terms. Find the Code of Conduct in the `CODE-OF-CONDUCT.md` file on GitHub or in the Code of Conduct section of read the docs.

If you have a problem using Octopod or see a possible improvement, open an issue in the GitHub issue tracker. Please be as specific as you can.

If you see an open issue you'd like to be fixed, take a stab at it and open a PR!

4.6.2 Pull Requests

To create a PR against this library, please fork the project and work from there.

Steps

1. Fork the project via the Fork button on Github
2. Clone the repo to your local disk.
3. Create a new branch for your PR.

```
git checkout -b my-awesome-new-feature
```

1. Install requirements (probably in a virtual environment)

```
virtualenv venv
source venv/bin/activate
pip install -r requirements-dev.txt
pip install -e .
```

1. Develop your feature
2. Submit a PR to main! Someone will review your code and merge your code into main when it is approved.

PR Checklist

- Ensure your code has followed the Style Guidelines below
- Run the linter on your code

```
source venv/bin/activate
flake8 octopod tests
```

- Make sure you have written unittests where appropriate
- Make sure the unittests pass

```
source venv/bin/activate
pytest -v
```

- Update the docs where appropriate. You can rebuild them with the commands below.

```
cd docs/
make html
open build/html/index.html
```

- Update the CHANGELOG

Style Guidelines

For the most part, this library follows PEP8 with a couple of exceptions.

- Indent with 4 spaces
- Lines can be 100 characters long
- Docstrings should be `numpy style` docstrings.
- Your code should be Python 3 compatible
- When in doubt, follow the style of the existing code
- We prefer single quotes for one-line strings unless using double quotes allows us to avoid escaping internal single quotes.

4.7 Contributor Covenant Code of Conduct

4.7.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

4.7.2 Our Standards

Examples of behavior that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologizing to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behavior include:

- The use of sexualized language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

4.7.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behavior and will take appropriate and fair corrective action in response to any behavior that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

4.7.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

4.7.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported to the community leaders responsible for enforcement by submitting this [anonymous form](#) or by sending an email to opensource@shoprunner.com. All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

4.7.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

1. Correction

Community Impact: Use of inappropriate language or other behavior deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behavior was inappropriate. A public apology may be requested.

2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behavior. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behavior.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behavior, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the project community.

4.7.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 2.0, available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

For answers to common questions about this code of conduct, see the FAQ at <https://www.contributor-covenant.org/faq>. Translations are available at <https://www.contributor-covenant.org/translations>.

PYTHON MODULE INDEX

O

`octopod.dataloader`, 9

M

module

 octopod.dataloader, 9

MultiDatasetLoader (*class in octopod.dataloader*),

 9

O

octopod.dataloader

 module, 9