

---

**dlproject**

***Release 0.1.1***

**Benjamin D. Killeen**

**Jul 08, 2021**



**CONTENTS:**

<b>1</b>	<b>Deep Learning Project Template</b>	<b>1</b>
1.1	Description . . . . .	1
1.2	Installation . . . . .	1
1.3	Usage . . . . .	2
<b>2</b>	<b>dlproject package</b>	<b>3</b>
2.1	Subpackages . . . . .	3
2.2	Submodules . . . . .	3
2.3	dlproject.experiments module . . . . .	3
2.4	dlproject.logging module . . . . .	3
2.5	dlproject.utils module . . . . .	3
2.6	Module contents . . . . .	3
<b>3</b>	<b>Indices and tables</b>	<b>5</b>



# DEEP LEARNING PROJECT TEMPLATE

*The opinionated deep learning template.*

## 1.1 Description

dlproject believes three things.

1. All code should be documented.
2. All experiments should be logged.
3. Configs are better than constants.

## 1.2 Installation

These instructions assume you are using a linux machine with at least one GPU (CUDA 11.1).

1. Create a new repository using this template and change to the root directory. For example,

```
git clone git@github.com:benjamindkilleen/dlproject.git
cd dlproject
```

2. Install dependencies using either [Anaconda](#) (preferred) or Pip:

- **Anaconda:** modify `environment.yml` to suit your needs. Then run:

```
conda env create -f environment.yml
conda activate dlproject
```

This will create a new environment with the project installed as an edit-able package.

- **Pip:** Install [Pytorch](#) to ensure GPU available. Then:

```
pip install -r requirements.txt
pip install -e .
```

## 1.3 Usage

The project is separated into “experiments,” which are just different main functions. Use the `experiment` group parameter to change which experiment is running. For example:

```
python main.py experiment=mnist
```

The results are then neatly sorted into the newly-created `results` directory (ignored by default). This is important for reproducibility, utilizing Hydra’s automatic logging and config storage.

### 1.3.1 Documentation

Documentation and tutorials for `dlproject` are available [here](#). You should document your code as you go. If you use Visual Studio Code, [this](#) is an extension which will create Google style docstrings automatically.

To build the docstrings you write into a local static web-page, run

```
pip install -r docs/requirements.txt
sphinx-apidoc -f -o docs/source dlproject
cd docs
make html
```

And open `/docs/build/html/index.html` in your browser.

### 1.3.2 Citation

```
@article{YourName,
  title={Your Title},
  author={Your team},
  journal={Location},
  year={Year}
}
```

## DLPROJECT PACKAGE

### 2.1 Subpackages

#### 2.1.1 `dlproject.datasets` package

Submodules

`dlproject.datasets.mnist` module

Module contents

#### 2.1.2 `dlproject.models` package

Submodules

`dlproject.models.classifier` module

Module contents

### 2.2 Submodules

#### 2.3 `dlproject.experiments` module

#### 2.4 `dlproject.logging` module

#### 2.5 `dlproject.utils` module

#### 2.6 Module contents





## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`