From Notebook to Production Code

Christopher Messier

General Assembly

Washington, DC

April 2, 2018

Overview

1. Introduction
2. Code Conversion
3. Build File
4. Command Line Interface
5. Progress Bar

What’s wrong with a Notebook?

An important tool, but have disadvantages

- Non-executable
- No returns
- JavaScript Bloat (*ewwww*)
- *slow*

What won't be covered

There's limited time available so we'll be skipping over...

- Optimization
- Unit Testing
- Packaging
- So much more...

Getting Started

We'll be working with the notebook found here:

`github.com/messiest/from-notebook-to-production-code`

- What it is:
  - A quick introduction to `PyTorch`
  - Implementation of the LeNet
  - Convolutional Neural Network on *MNIST*

Converting Your Code

Exporting your code to a Python (`.py`) file

- File → Download As → Python (.py)
- Move file from Downloads to the Project directory
- Clean out notebook ”artifacts”

Build File

Install all packages required for a project

- `requirements.txt` - list of all packages
- Could be either a Python (`.py`) or Bash (`.sh`) script
- Ensures that there is a common build for all users

Command Line Interface

Interacting with your code through the command line

- Makes your code more flexible
- Gives greater control over code behavior
- Provides a common interface for

argparse

Part of the Python Standard Library (included w/ Python)

- Allows you to define custom arguments
- Provides help read-outs by default
- Easy to use, and descriptive

Progress Bar

Provides a simple way to track your program's progress

- Estimate how long your program will run
- Track values of important variables
- Provides a professional look/feel

tqdm

Provides a wrapper to output progress to command line

- Wraps an iterable, and automatically writes output
- Provides a way to customize the progress bar

Thank You