2022-05-25

TensorFlow and Docstring

Contribute to the Tensorflow documentation

- Tensorflow welcomes documentation contributions
- Documentation on tensorflow.org falls into the following categories:
  - API reference.
    - generated from docstring in the TensorFlow source code
  - Narrative documentation
    - Tutorials, guide, and other writing thats not of the Tensorflow code. This documentation is in the tensorflow/docs GitHub repo
  - community translation
    - Guides and tutorials translated by community.
- Some TensorFlow projects keep documentation near code in a seperate repo, usually in docs/ directory

Version and branches

- API reference version defaults to the latest stable binary
- default TensorFlow package is fom the Stable braneh rX.x in the main tensorflow/tensorflow repo.

Build API docs

- Step is not required to edit or preview API docstring, only to generate the HTML used on tensorflow.org

Python Reference.

- A script uses the installed TensorFlow package to generate docs and only works for Tensorflow 2.x.

Narrative Documentation

- written as Markdown files & interactives (jupyter Notebooks)
- Narrative docs on tensorflow.org are built from
- Notebooks can be run in browse

Set up a local Git repo

- For mutiline/file or more complex updates

Simple changes can be made using Github's web based file editor

Fork tensorflow/docs repo

- on tensorflow/docs github click fork button

Clone your repo

- download a copy remote username/docs on local machine

Add an upstream repo to keep up to date (optional)

Interactive Notebooks

- Google colaboratory is a hosted notebook

Notebook Formatting

- A notebook formatting tool makes jupyter notebook source diffs consistent and easier to review
- Tensorflow docs projects, notebooks without output cells are published as is

Edit in Colab

- Google Colab environment
- double-click cells to edit text & code blocks
- Text cells use markdown & should follow the tensorflow docs style guide.

Contribute to the TensorFlow API documentation

Testable docstring

- TensorFlow uses DocTest to test code snippet in Python docstring
- snippet must be executable Python code

Make code testable with DocTest

- Docstrings use backticks (```) to identify code.
  - Remove the backticks & use the left brackets (>>>) in front of each line
  - Add a new line to seperate DocTest snippets from Markdown text to render properly on tensorflow.org

Customization

- Tensorflow uses a few customizations to builtin doctest logic.
- Doesn't compare float values as text
- Float values are extracted from the text and compared using allclose with liberal atol & rtol tolerances.
  - Clearer docs - Authors dont need to include all decimal places
  - More Robust test - Numerical change in the Underlying implementation should never cause a doctest to fail

- Only checks output if the author includes otpt for a line

Docstring Considerations

- Overall: The goal of doctest is to provide documentation, and confirm that the documentation works. this is different from unit Testing
  - Keep examples simple
  - Avoid long or complicated outputs
  - use round numbers if possible