ONNX Git Hands-on

Chin Huang, Cognitive OpenTech

Data and AI
Open Source Dojo

Outline

- Unit 1: Git overview and setup
- Unit 2: The onnx-dojo repository
- Unit 3: Create branch and commit
- Unit 4: Create and merge pull request

Unit1: Git overview and setup

- **Goal of unit 1: At the end of unit 1, you will learn git, github, repo, and setup your git environment for remaining labs**

- Prereqs: introduction, details, install, and config videos
- Git is an open-source version control system
- Git stores file changes efficiently and ensures file integrity
- Git is a command-line tool
- Github is where developers store their projects and network with others
- A repository (abbreviated “repo”) is a location where all the files for a particular project are stored.

Unit1: Git overview and setup

- Step 1.1: Install git
  - Ubuntu: *$ sudo apt install git*
  - OS X: *$ brew install git*
  - Verify: *$ git --help*
- Step 1.2: Configure git (alternatively using .gitconfig)
  - User name and email
    - *$ git config --global user.name "John Doe"*
    - *$ git config --global user.email johndoe@example.com*
  - Setup a default text editor when git needs you to type in a message
    - *$ git config --global core.editor vi*
  - Setup a push default option to simple
    - *$ git config --global push.default simple*
  - Check settings: *$ git config --list*

Unit 2: Fork and clone ONNX-Dojo repo

- **Goal of unit 2: At the end of unit 2, you will learn the onnx-dojo repo and how to create a fork, clone a repo, and setup an upstream**

- Quick overview: just about everything for the ONNX-Dojo training (https://github.com/chinhuang007/onnx-dojo)
  - Presentations under the docs folder
  - Labs under lab_xxx folders with all the code and files to complete the exercises
  - Participants under profiles folder

Unit 2: Fork and clone ONNX-Dojo repo

- Step 2.1: fork the repo
  - “Fork” is when you create a new project based off of another project that already exists. It makes a copy the project in the cloud so you can easily make changes, and later create and merge pull requests into the upstream project
  - In a browser, go to https://github.com/chinhuang007/onnx-dojo, and click ‘Fork’ on the upper right hand corner. When prompted, make the fork go to your user account.
  - Verify: Check in browser, you should have a repository https://github.com/<user-name>/onnx-dojo

Unit 2: Fork and clone ONNX-Dojo repo

- Step 2.2: clone the repo
  - “git clone” clones a repository into a new directory as a local repository, for local changes
  - *$ git clone https://github.com/<user-name>/onnx-dojo.git*
  - Verify: a folder ‘onnx-dojo’ is created
  - Verify: cd to ‘onnx-dojo’ and *$ git remote -v* should return 2 entries with URLs for your remote repository origin
- Step 2.3: setup upstream
  - “upstream” is from where you clone the repository so in our case is https://github.com/chinhuang007/onnx-dojo
  - *$ git remote add upstream https://github.com/chinhuang007/onnx-dojo.git*
  - Verify: in ‘onnx-dojo’ folder *$ git remote –v* should return 4 entries with 2 each for origin and upstream

Unit 3: Create branch and commit

- **Goal of unit 3: At the end of unit 3, you will learn how to create a branch, make changes, and push a commit**

- Quick overview: You are going to add a short intro doc about yourself under the profiles folder. A reference is https://github.com/chinhuang007/onnx-dojo/blob/master/profiles/ChinHuang.md. Please include at least your name and email address. Anything else is a plus ☺

Unit 3: Create branch and commit

- Step 3.1: create a branch
  - Check the current branch first *$ git branch* should show just the master branch
  - To make changes and later create commits, we need to create a local branch to work on.
  - *$ git checkout –b add-your-name-profile* (for ex. add-chin-huang-profile)
  - Verify: *$ git branch* should show two branches now and the active one with * mark
- Quick tips:
  - Always work on a new branch and keep master clean
  - Use one branch to solve one problem
  - Limit the size of changes in a branch
  - Never work directly on or commit to master!

Unit 3: Create branch and commit

- Step 3.2: create your profile
  - Create a file under profiles folder. You could use .md or simple text file.
  - Check your branch status
    - *$ git status* should show you are on your branch with an untracked file
- Step 3.3: create a commit
  - First need to add the changes to be committed
    - *$ git add your-file-name*
    - *$ git status* should show a new file as changes to be committed
  - Next create a commit
    - *$ git commit* should open the editor you specified in step 1.2.
    - The first line is the commit title, such as “Add John Doe”.
    - Put an empty line and the text below will be the commit message, describing why the commit exists
    - Once done exit the editor with a normal save and edit

Unit 3: Create branch and commit

  - Check your branch status
    - *$ git status* should show you are on your branch with nothing to commit
- Step 3.4: push up a commit
  - Now you need to push the commit up to the remote repo.
    - *$ git push origin add-your-name-profile*
    - Enter your git user name and password when prompted
  - Check your commit is pushed up and ready for a pull request
    - In a browser, go to https://github.com/chinhuang007/onnx-dojo and your branch should appear as ready for a pull request
    - Next go to https://github.com/<user-rname>/onnx-dojo and click on Branch: master and switch to "*add-your-name-profile" branch*
    - Navigate to profiles folder and your newly added file should be there

Unit 4: Create and merge pull request

- **Goal of unit 4: At the end of unit 4, you will learn how to create, review and merge a pull request**

- A pull request tells other community members about the changes you have pushed to a repository
- Other contributors can review your proposed changes, add review comments, contribute to the pull request discussion, and even add commits to the pull request.
- After everyone is happy with the proposed changes, the pull request will be merged into the target branch.
- Step 4.1: create a pull request
  - In a browser, go to https://github.com/chinhuang007/onnx-dojo and click “compare & pull request” button for your branch
  - You could add more detailed message, screen shots, links to issues and other PRs. Let’s add a person mention so he/she will get an email notification for the change.

Unit 4: Create and merge pull request

  - Type ‘@chinhuang007 Check this out” in the text field.
  - Click “Create pull request” button.
  - Verify: The pull request is created. Notification is sent to the person you specified earlier. If there is Continuous Integration (CI) configured for the repo, CI tasks will be run (not in this example).
  - Verify: You should see “This branch has no conflicts with the base branch”.

- Step 4.2: review a pull request (instructors walk-through)
  - In a browser, go to https://github.com/chinhuang007/onnx-dojo and click “Pull Requests” tab.
  - Find the pull request with title “Add John Doe”
  - Associate with an issue (need permissions)
  - Review the changes and run tests if needed
  - Leave a comment
  - Provide final review and approval (need permissions). Other options: comment and request changes
  - Verify: The pull request has one approval.
  - Verify: The “merge pull request” button is green for the users with merge permissions.

Unit 4: Create and merge pull request

- Step 4.3: merge a pull request (instructor walk-through)
  - Click “Merge Pull Request”
  - Click “Confirm Merge”
  - Verify: The pull request status is changed from “Open” to “Closed”
  - Verify: The associated issue is also changed from “Open” to “Closed”
  - Verify: The repo is updated with the changes.
- Step 4.4: delete working branch and rebase master (optional)
  - Once a pull request is merged, the working branch can be deleted.
    - *$ git checkout master*
    - *$ git branch -D add-your-name-profile*
    - *$ git branch* should show master only
    - You can delete the remote branch from UI, either from upstream or your repo
  - Rebase your master branch so you are in sync with upstream master.
    - *$ git fetch upstream*
    - *$ git rebase upstream/master*

Unit 4: Create and merge pull request

- *$ git push origin master*
- Enter your git user name and password when prompted
- Your remote repo master should be updated now.
- Verify: go to https://github.com/<user-name>/onnx-dojo and check your master branch is even with upstream master
- Verify: navigate through https://github.com/<user-name>/onnx-dojo to see the latest contents
- Now you are good to go, working on next PR!