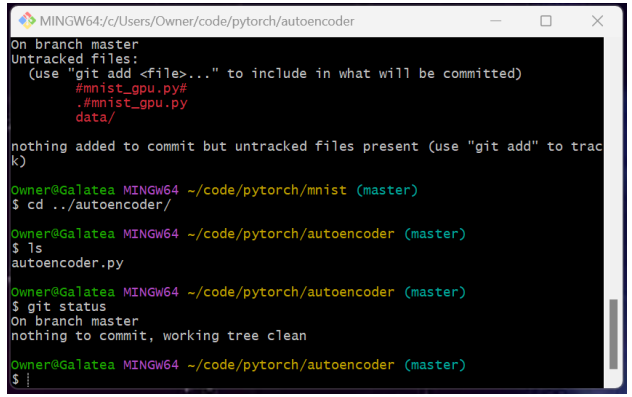
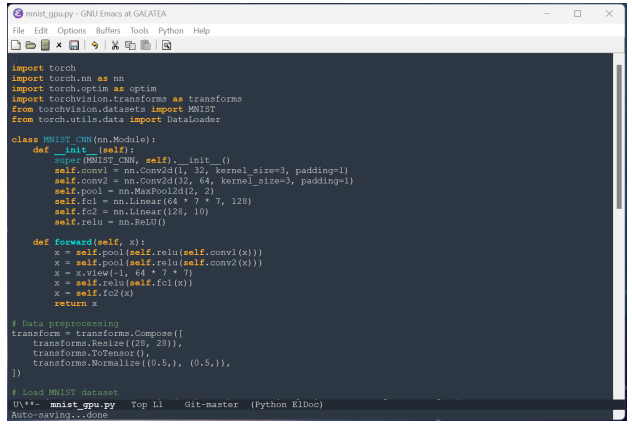


CTD Intro Week 7

Introduction to git and IDEs

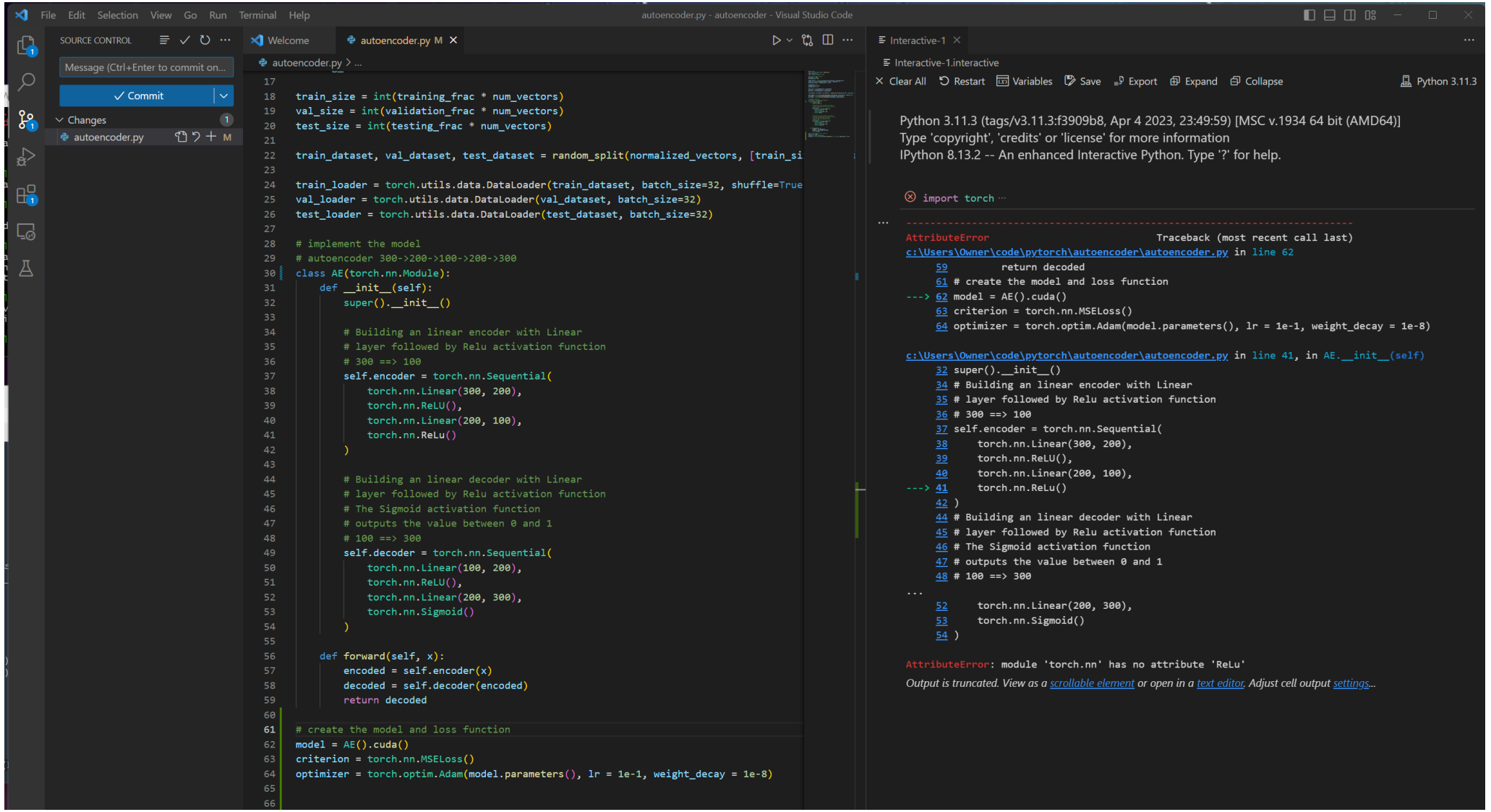
GitHub



Text Editors and IDEs

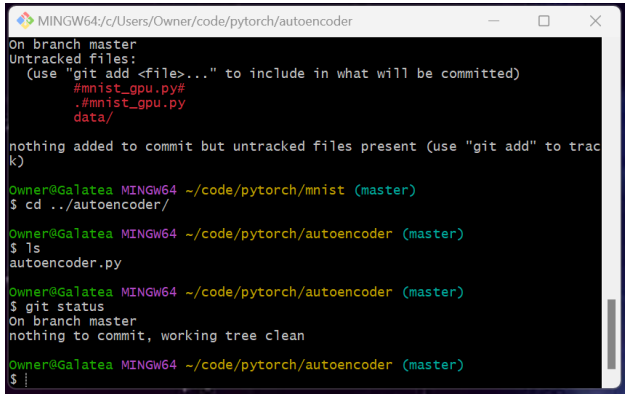
- Enter program text with formatting assistance
- IDE – Integrated Development Environment
  - Versioning system
  - Interactive windows
  - Debugger
  - Coding hints
- Lots of options
  - If you don’t already have a favorite, Vscode is recommended

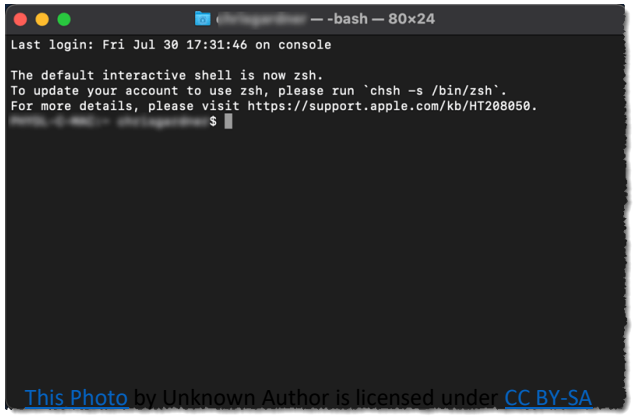
Vscode IDE

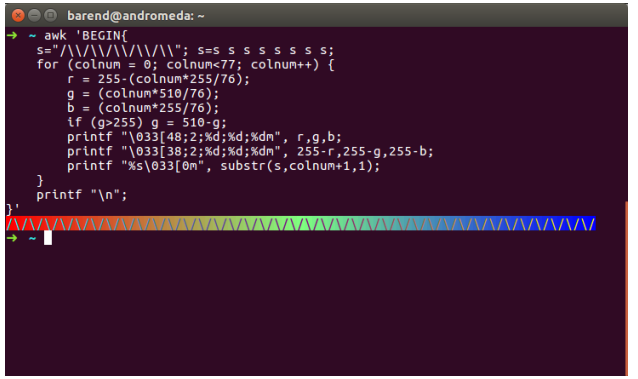

File Edit Selection View Go Run Terminal Help
autoencoder.py - autoencoder - Visual Studio Code
SOURCE CONTROL
Message (Ctrl+Enter to commit on...
Commit
Changes 1
autoencoder.py M
Welcome
autoencoder.py M
autoencoder.py > ...
train_size = int(training_frac * num_vectors)
val_size = int(validation_frac * num_vectors)
test_size = int(testing_frac * num_vectors)
train_dataset, val_dataset, test_dataset = random_split(normalized_vectors, [train_si
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=32, shuffle=True
val_loader = torch.utils.data.DataLoader(val_dataset, batch_size=32)
test_loader = torch.utils.data.DataLoader(test_dataset, batch_size=32)
# implement the model
# autoencoder 300->200->100->200->300
class AE(torch.nn.Module):
def __init__(self):
super().__init__()
# Building an linear encoder with Linear
# layer followed by Relu activation function
# 300 ==> 100
self.encoder = torch.nn.Sequential(
torch.nn.Linear(300, 200),
torch.nn.ReLU(),
torch.nn.Linear(200, 100),
torch.nn.ReLu()
)
# Building an linear decoder with Linear
# layer followed by Relu activation function
# The Sigmoid activation function
# outputs the value between 0 and 1
# 100 ==> 300
self.decoder = torch.nn.Sequential(
torch.nn.Linear(100, 200),
torch.nn.ReLU(),
torch.nn.Linear(200, 300),
torch.nn.Sigmoid()
)
def forward(self, x):
encoded = self.encoder(x)
decoded = self.decoder(encoded)
return decoded
# create the model and loss function
model = AE().cuda()
criterion = torch.nn.MSELoss()
optimizer = torch.optim.Adam(model.parameters(), lr = 1e-1, weight_decay = 1e-8)
Interactive-1
Interactive-1.interactive
Clear All Restart Variables Save Export Expand Collapse
Python 3.11.3
Python 3.11.3 (tags/v3.11.3:f3909b8, Apr 4 2023, 23:49:59) [MSC v.1934 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.13.2 -- An enhanced Interactive Python. Type '?' for help.
import torch
AttributeError Traceback (most recent call last)
c:\Users\Owner\code\pytorch\autoencoder\autoencoder.py in line 62
59 return decoded
61 # create the model and loss function
---> 62 model = AE().cuda()
63 criterion = torch.nn.MSELoss()
64 optimizer = torch.optim.Adam(model.parameters(), lr = 1e-1, weight_decay = 1e-8)
c:\Users\Owner\code\pytorch\autoencoder\autoencoder.py in line 41, in AE.__init__(self)
32 super().__init__()
34 # Building an linear encoder with Linear
35 # layer followed by Relu activation function
36 # 300 ==> 100
37 self.encoder = torch.nn.Sequential(
38 torch.nn.Linear(300, 200),
39 torch.nn.ReLU(),
40 torch.nn.Linear(200, 100),
---> 41 torch.nn.ReLu()
42)
44 # Building an linear decoder with Linear
45 # layer followed by Relu activation function
46 # The Sigmoid activation function
47 # outputs the value between 0 and 1
48 # 100 ==> 300
...
52 torch.nn.Linear(200, 300),
53 torch.nn.Sigmoid()
54)
AttributeError: module 'torch.nn' has no attribute 'ReLu'
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Command Line

- Convenient way to
  - Navigate, perform git operations, run programs…
- Mac and linux include terminal applications
- Windows provides powershell, but git bash is better
  - Git and git bash can be installed through vscode
- Commands
  - pwd – current directory
  - ls – list files, ls –lah (more details)
  - cd – change directory (~ is a shortcut for your home directory)
  - cp – copy, mv – rename
  - rm – remove/delete
  - diff – differences between two text files
  - The first name on the line is a program or script which will run
  - man <command> to get documentation







Setting up git

- Windows, use git bash
  - Can be done through vscode
- Mac
  - May already be there (run ‘which git’ to find out)
  - Use homebrew (brew install git)
- Linux (Ubuntu)
  - May already be there (which git)
  - sudo apt install git
- Check the version (git –version) should be 2.28 or greater
- Setting Up Git | The Odin Project

Versioning and intro to git

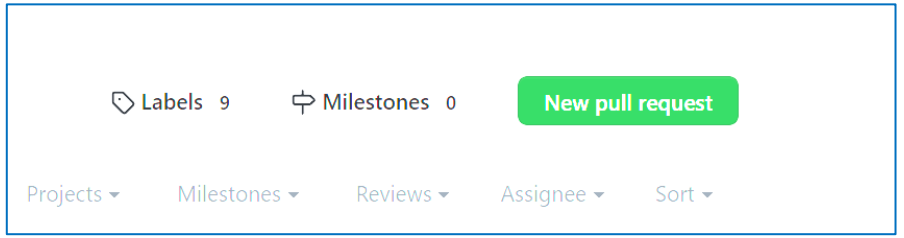
- What is a versioning system?
- History
- Git and github workflow
  - Very popular (e.g. 84% of fortune 100)
  - Github – central repository for collaboration (built on RoR!)
- Git features
  - Tracks content, not dates (SHA)
  - Clone entire repository
  - Easy branching and branch management
  - Pull requests for review and merging of changes
- Vscode works well with git (Microsoft owns github)

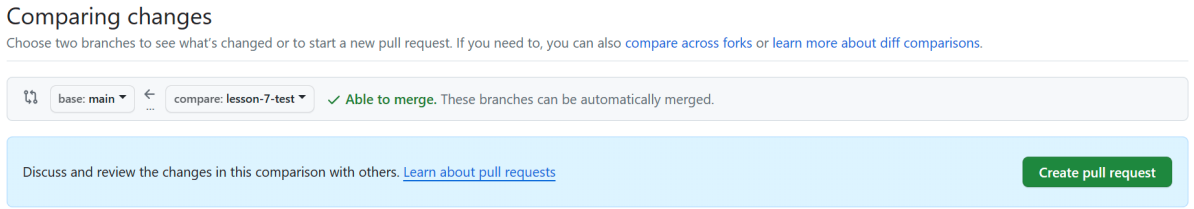
Git basics

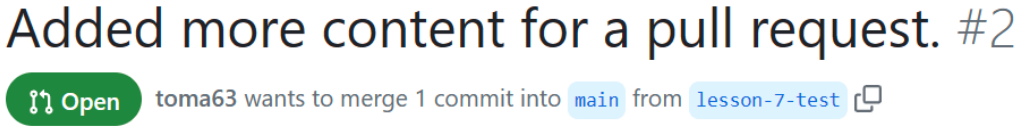
- Create a new public repository in your github account (e.g. *yourName-h*aumea)
  - New repository (github.com) – green button on the upper left in your dashboard view
- Clone the github repo to your local machine
  - git clone https://github.com/*YourGithubHandle/your-new-repository*.git (your forked repo)
  - This is run on your local command line in the directory (folder) where you put CTD repositories
    - E.g. ~/code-the-dream
- git init (set up a local repository – not needed if cloning)
- git status (which file are modified, etc.)
- git diff (what's changed)
- git log (all the commit log messages)
- git branch (what branches are there?, what's the current branch?)
- git checkout (change branches)
  - git checkout –b branch-name (create a new branch with current changes)
- git add (stage files for commit)
- git commit –m commit log message (opens editor if no –m)
- git push (pushes changes upstream e.g. to github)
- git pull (pull changes from upstream e.g. github)

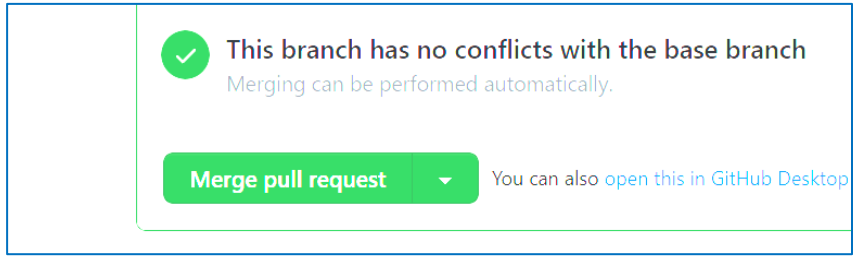
Pull Requests

- Standard workflow for making changes to a shared repository
- Allows your supervisor and peers to review and comment
- Flow:
  - Clone the repository
  - Make a new branch for your changes ‘git checkout –b lesson-7’
  - Make and validate your edits
  - Push your branch to github ‘git push’
    - may need `git push --set-upstream origin lesson-7` the first time
  - Create a pull request (PR) from your branch
  - Request reviews for your PR
    - You can push more commits to the pull request branch to address review feedback.
  - Merge your pull request when reviews are satisfied









Q & A and Demo