



# CTD Intro Week 7

Introduction to git and IDEs



# GitHub

```
MINGW64/c/Users/Owner/code/pytorch/autoencoder
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        mnist_gpu.py
        mnist_gpu.py
        data/

nothing added to commit but untracked files present (use "git add" to track)

Owner@Galatea MINGW64 ~/code/pytorch/mnist (master)
$ cd ../autoencoder/

Owner@Galatea MINGW64 ~/code/pytorch/autoencoder (master)
$ ls
autoencoder.py

Owner@Galatea MINGW64 ~/code/pytorch/autoencoder (master)
$ git status
On branch master
nothing to commit, working tree clean

Owner@Galatea MINGW64 ~/code/pytorch/autoencoder (master)
$
```

```
mnist_gpu.py - GPU traces at GALATEA
File Edit Options Buffers Tools Python Help

import torch
import torch.nn as nn
import torch.optim as optim
import torchvision.transforms as transforms
from torchvision.datasets import MNIST
from torch.utils.data import DataLoader

class MNIST_CNN(nn.Module):
    def __init__(self):
        super(MNIST_CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=3, padding=1)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=3, padding=1)
        self.pool = nn.MaxPool2d(2, 2)
        self.fc1 = nn.Linear(64 * 7 * 7, 128)
        self.fc2 = nn.Linear(128, 10)
        self.relu = nn.ReLU()

    def forward(self, x):
        x = self.pool(self.relu(self.conv1(x)))
        x = self.pool(self.relu(self.conv2(x)))
        x = x.view(-1, 64 * 7 * 7)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x

# Data preprocessing
transform = transforms.Compose([
    transforms.Resize((28, 28)),
    transforms.ToTensor(),
    transforms.Normalize((0.5), (0.5)),
])

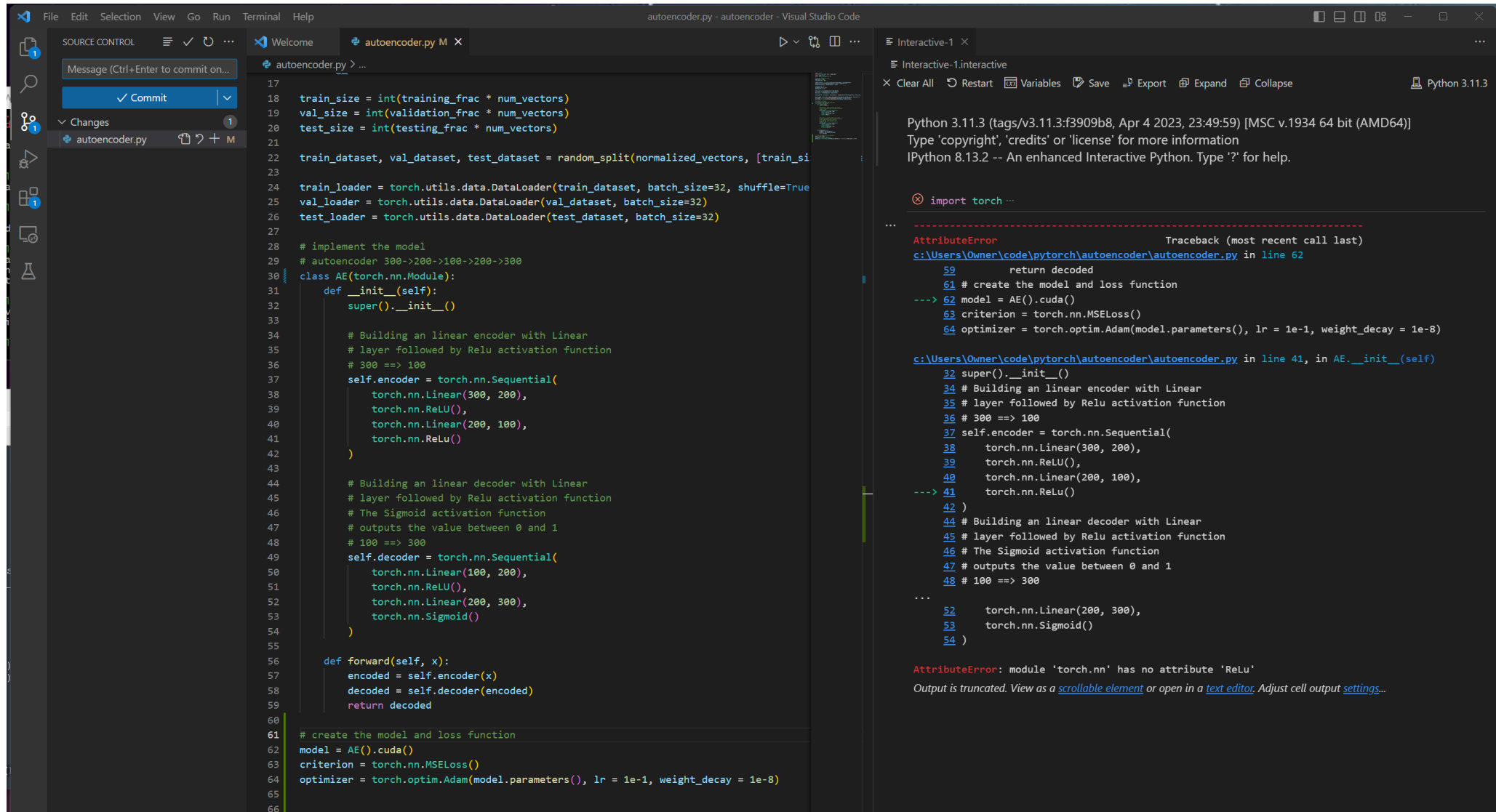
# Load MNIST dataset
U** mnist_gpu.py Top LI Git-master (Python 3.10.0)
```

# 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



The screenshot displays the Visual Studio Code (VS Code) IDE interface. The main editor window shows a Python file named `autoencoder.py` with the following code:

```
17
18 train_size = int(training_frac * num_vectors)
19 val_size = int(validation_frac * num_vectors)
20 test_size = int(testing_frac * num_vectors)
21
22 train_dataset, val_dataset, test_dataset = random_split(normalized_vectors, [train_si
23
24 train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=32, shuffle=True
25 val_loader = torch.utils.data.DataLoader(val_dataset, batch_size=32)
26 test_loader = torch.utils.data.DataLoader(test_dataset, batch_size=32)
27
28 # implement the model
29 # autoencoder 300->200->100->200->300
30 class AE(torch.nn.Module):
31     def __init__(self):
32         super().__init__()
33
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         )
43
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
49         self.decoder = torch.nn.Sequential(
50             torch.nn.Linear(100, 200),
51             torch.nn.ReLU(),
52             torch.nn.Linear(200, 300),
53             torch.nn.Sigmoid()
54         )
55
56     def forward(self, x):
57         encoded = self.encoder(x)
58         decoded = self.decoder(encoded)
59         return decoded
60
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)
65
66
```

The left sidebar shows the 'Source Control' panel with a 'Commit' button and a 'Changes' section. The right sidebar shows the 'Interactive-1' window with the following output:

```
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 ...

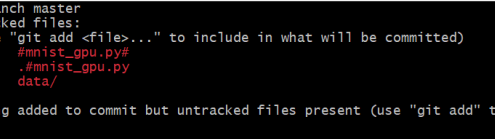
AttributeError                                Traceback (most recent call last)
c:\Users\Owner\code\pytorch\autoencoder\autoencoder.py in line 62
    59         return decoded
    61 # create the model and loss function
--> 62 model = AE().cuda()
    63 criterion = torch.nn.MSELoss()
    64 optimizer = 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
    49
    50     torch.nn.Linear(200, 300),
    51     torch.nn.Sigmoid()
    52 )
    53
    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



```

MINGW64:/c/Users/Owner/c/pytorch/autoencoder
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        #mnist_gpu.py#
        .#mnist_gpu.py
        data/

nothing added to commit but untracked files present (use "git add" to track)

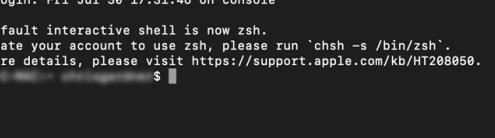
Owner@Galatea MINGW64 ~/code/pytorch/mnist (master)
$ cd ../autoencoder/

Owner@Galatea MINGW64 ~/code/pytorch/autoencoder (master)
$ ls
autoencoder.py

Owner@Galatea MINGW64 ~/code/pytorch/autoencoder (master)
$ git status
On branch master
nothing to commit, working tree clean

Owner@Galatea MINGW64 ~/code/pytorch/autoencoder (master)
$

```



A terminal window titled "Terminal" with a window size of "80x24". The prompt is "--bash--". The output shows the last login time as "Fri Jul 30 17:31:46 on console". The message states: "The default interactive shell is now zsh. To update your account to use zsh, please run 'chsh -s /bin/zsh'. For more details, please visit https://support.apple.com/kb/HT208850." The user has entered the command "chsh" and the prompt is now "s\$".

```
Terminal --bash-- 80x24
Last login: Fri Jul 30 17:31:46 on console

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208850.

chsh
s$
```

```
➔ ~ awk 'BEGIN{  
    s="/\\//\\\\//\\\\\\\\\\";  
    for (colnum = 0; colnum<77; colnum++) {  
        r = 255-(colnum*255/76);  
        g = (colnum*510/76);  
        b = (colnum*255/76);  
        if (g>255) g = 510-g;  
        printf "\\033[48;2;%d;%d;%dm", r,g,b;  
        printf "\\033[38;2;%d;%d;%dm", 255-r,255-g,255-b;  
        printf "%s\\033[0m", substr(s,colnum+1,1);  
    }  
    printf "\\n";  
}',  
➔ ~ █
```

# 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)



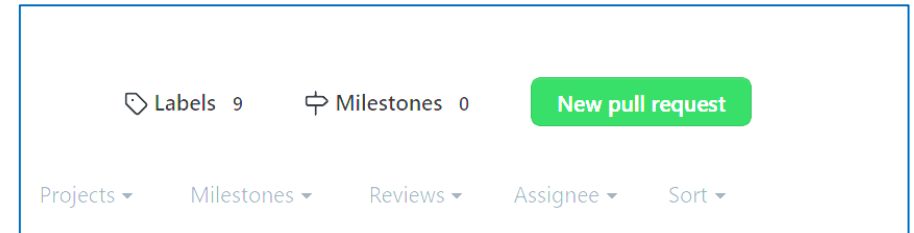
# Git basics

- Create a new public repository in your github account (e.g. *yourName-haumea*)
  - [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)



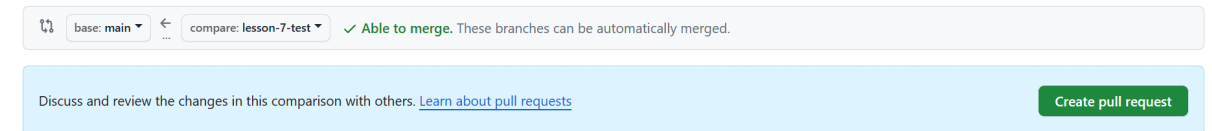
# Pull Requests

- 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



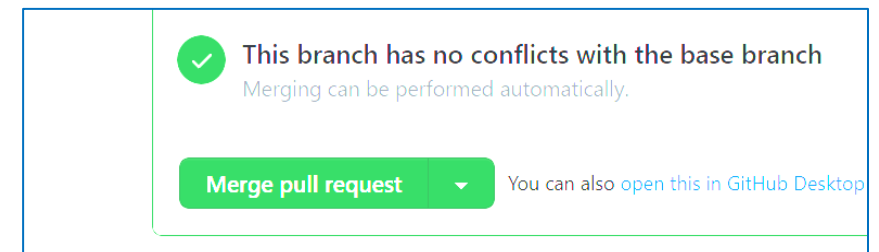
## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#) or [learn more about diff comparisons](#).



Added more content for a pull request. #2

Open toma63 wants to merge 1 commit into `main` from `lesson-7-test`





# Q & A and Demo

---

