PyTorch Bootcamp

Machine Learning and Deep Learning

A. A. 2024/25

Teaching Assistants: *Davide Buoso, Arda Eren Dogru*
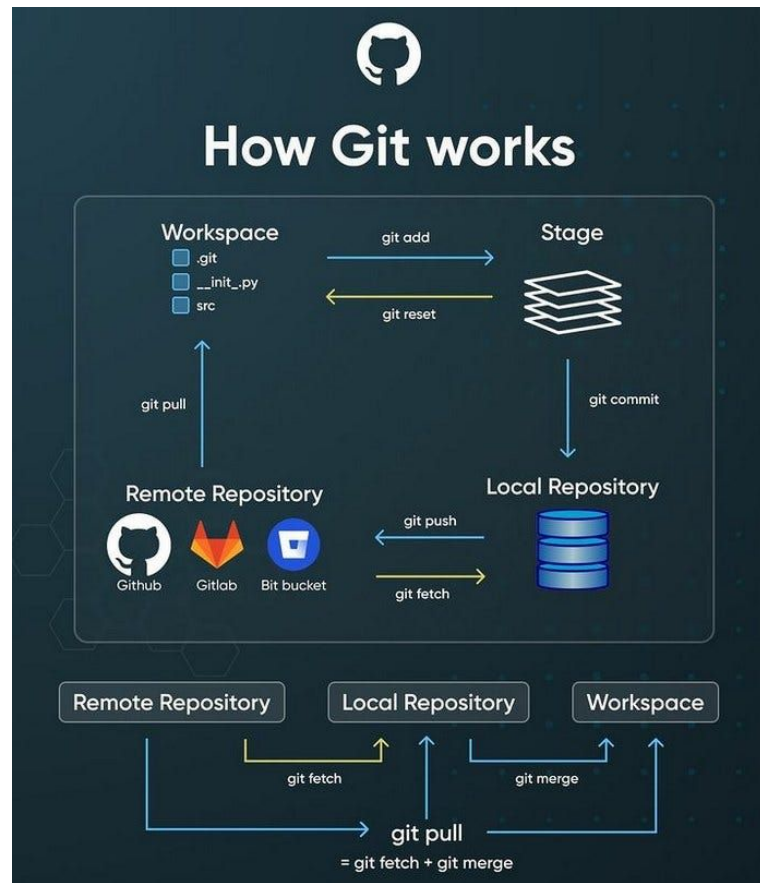*Slide credits:* *Chiara Plizzari, Gabriele Rosi*

How to setup a project from scratch

Lesson 3

Github

- Cloud-based platform
- Born for collaboration and version control

Repositories are cloud-saved folders saving remotely all your files

Github

Main commands:

- **Clone**: copies a repository to your machine (execute the first time)
- **Status**: tells us about the status of our repo
- **Add**: adds the local changes to the "to send remotely" files
- **Commit**: prepares "the package" of changes to send (and let you add a message, representing what you did)
- **Push**: Really pushes everything to your remote repo

- **Pull**: sync your local changes with the changes happened in the remote repo (useful to bring in your local folder the changes a team member pushed)

https://docs.github.com/en/get-started

4

Github

Typical Workflow

```
git clone <repo>


For every big change you make:
    … local changes …
    git add .
    git commit -m 'Explain message'


    git push
```
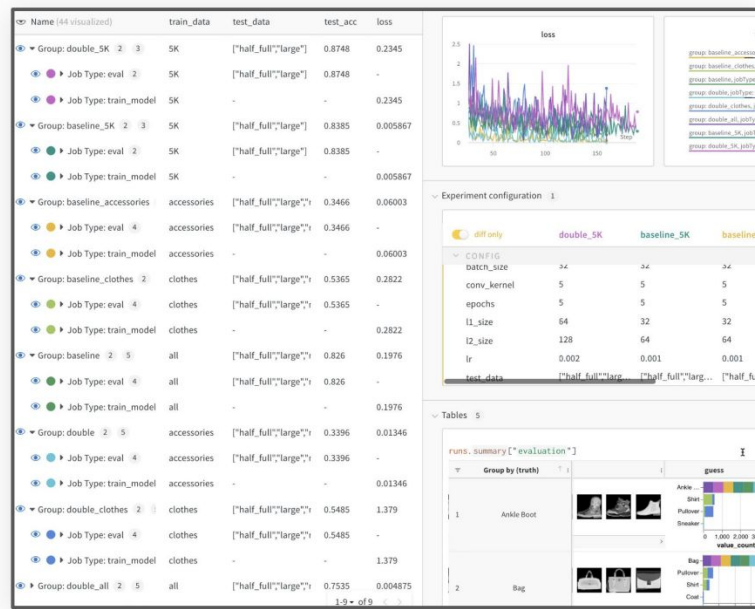
How to keep track of your results?

Wandb: record all your results

Get started in 60 seconds
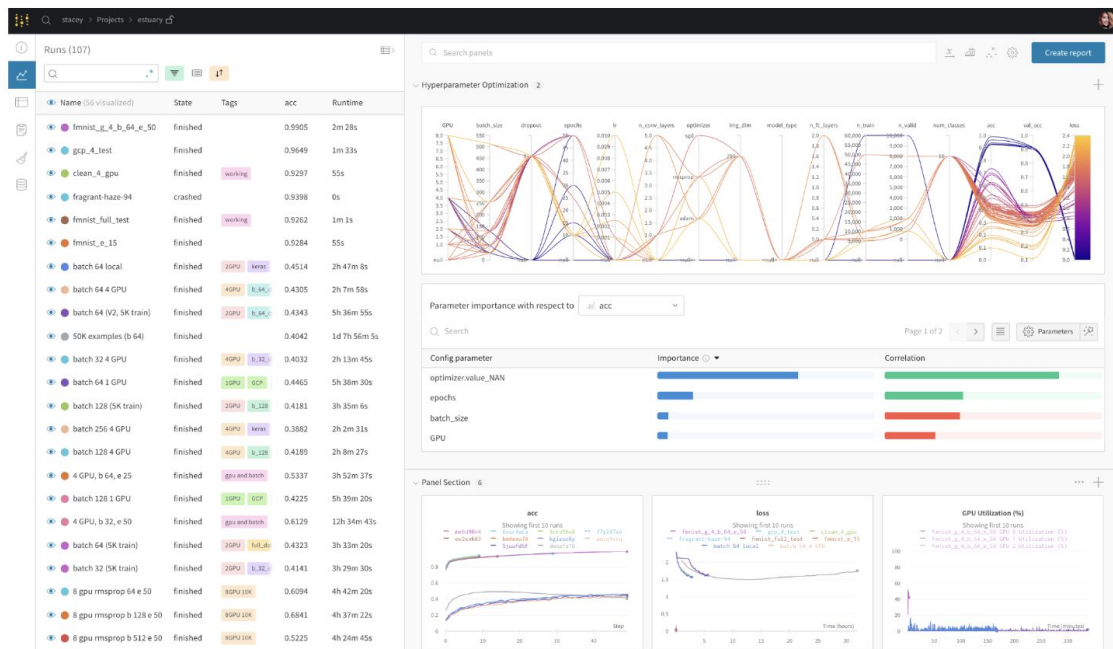
```
!pip install wandb      # Install W&B

wandb.init()            # Start experiment
wandb.log(metrics)      # Log metrics + more!
```

Never lose your progress

Save everything you need to debug, compare and reproduce your models — architecture, hyperparameters, weights, model predictions, GPU usage, git commits, and even datasets — with a few lines of code.

Track your model training
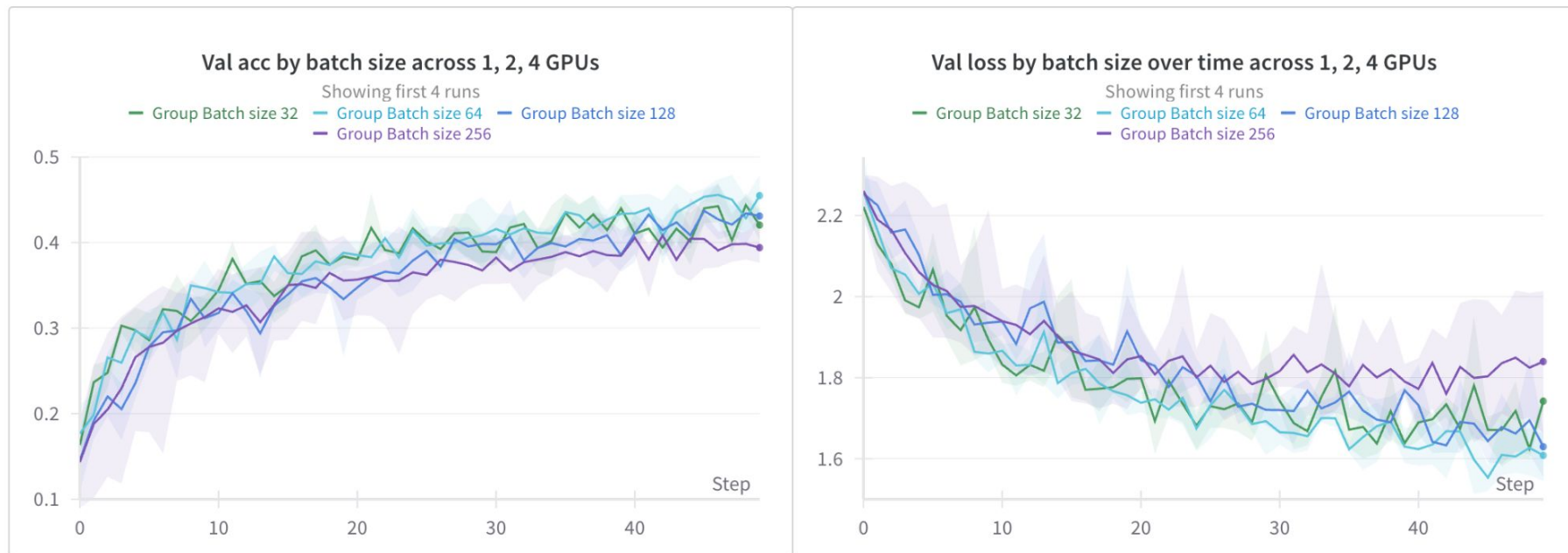
```python
import wandb

# 1. Start a W&B run
wandb.init(project='gpt3')

# 2. Save model inputs and hyperparameters
config = wandb.config
config.learning_rate = 0.01

# Model training code here

# 3. Log metrics over time to visualize performance
for i in range (10):
    wandb.log({"loss": loss})
```
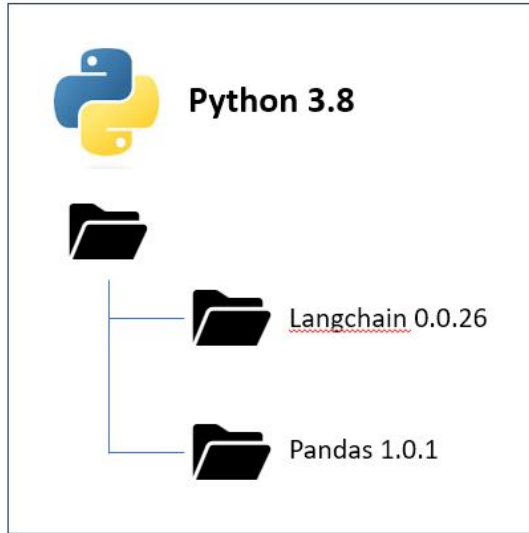
Track your model training

Virtual Envs and Pip Package Manager

System's Python
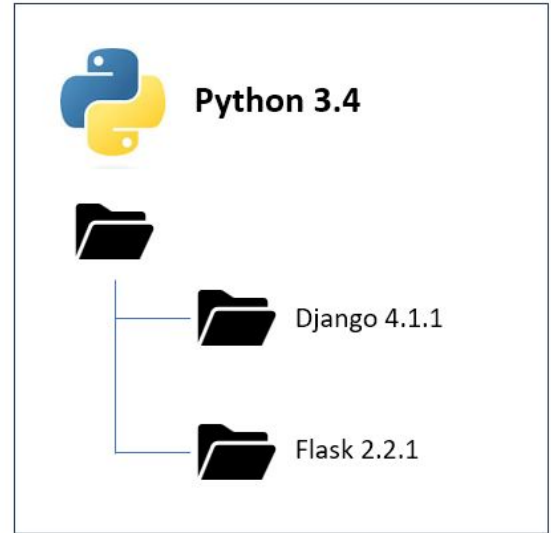
Colab special commands

! <command>: executes a bash command (e.g ls, cp, mv, git..)

```
!git clone https://github.com/lambdavi/mldl_project_skeleton.git
```

% <command>: executes a special command (e.g. cd)

```
%cd mldl_project_skeleton
```
```
/content/mldl_project_skeleton
```

Goal of this lab:

Use Github, Wandb and Pip to setup a structured project.

Name	Last commit message
📁 checkpoints	initial structure
📁 data	initial structure
📁 dataset	initial structure
📁 models	initial structure
📁 utils	initial structure
📄 .gitignore	Initial commit
📄 README.md	Initial commit
📄 eval.py	initial structure
📄 requirements.txt	Create requirements.txt
📄 train.py	initial structure

Here you will save your saved models

Here you will save your dataloader/datase classes

Here you will save your actual dataset(s)

Here you will save your architectures (Pytorch)

Here you will save utils that support your code (e.g. scripts/visualization etc)
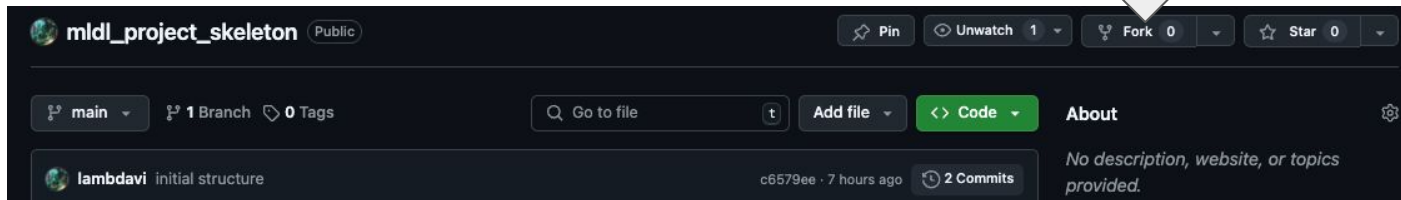
eval logic

file containing the pip packages you installed (for reproducibility)
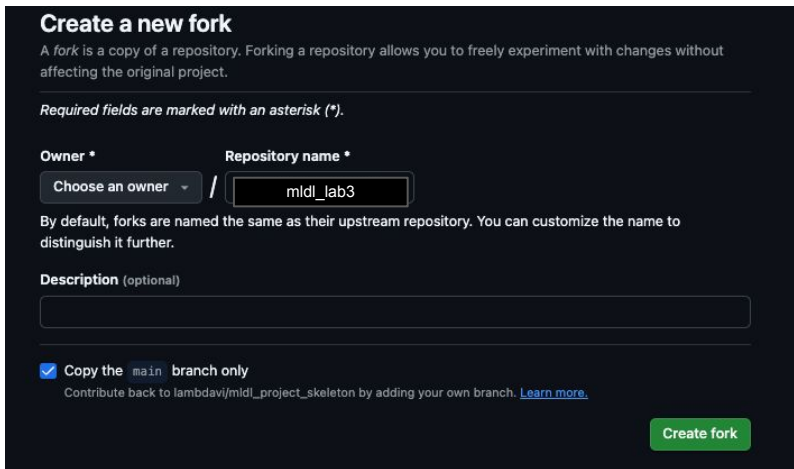
training logic

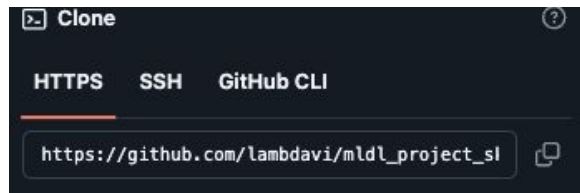https://github.com/lambdavi/mldl_project_skeleton

13

Github

Step 1: Fork the project

Step 2: name your repository
(e.g. mldl_lab3)

Step 3: Copy the link to the repo
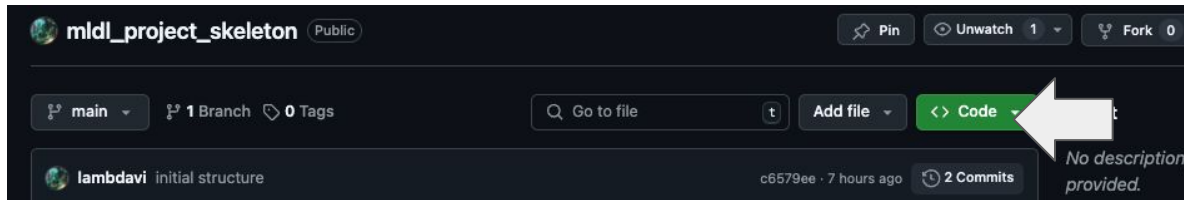
Step 4: Use CMD/Terminal to clone the repository you forked (you may have to install git if it says "command not found")

```
davidebuoso@MacBook-Pro-di-Davide ~ % git clone <repo>
```

Step 5: Open the folder on your favourite code editor (VSCode etc)

EXPLORER   ...

∨ MLDL_PROJECT_SKELETON
  > checkpoints
  > data
  > dataset
  > models
  > utils
  ◇ .gitignore
  🐍 eval.py
  ⓘ README.md
  ☰ requirements.txt   U
  🐍 train.py   M
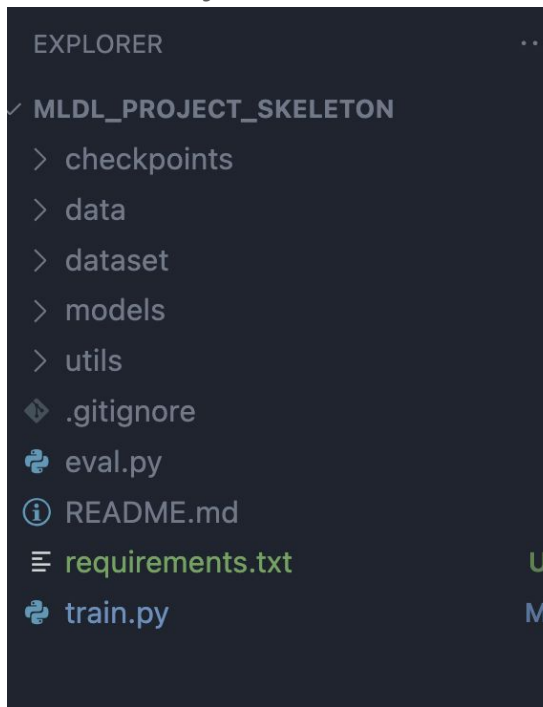
Step 6: Make the changes to the files, add new files etc

```
≡ requirements.txt U        🐍 train.py M       🐍 customnet.py U  ●

models >  🐍 customnet.py > ...
  1    from torch import nn
  2
  3    # Define the custom neural network
  4    class CustomNet(nn.Module):
  5        def __init__(self):
  6            super(CustomNet, self).__init__()
  7            # Define layers of the neural network
  8            self.conv1 = nn.Conv2d(3, 64, kernel_size=3, padding=1, stride=2)
  9            self.conv2 = nn.Conv2d(64, 128, kernel_size=3, padding=1, stride=2)
 10            self.conv3 = nn.Conv2d(128, 256, kernel_size=3, padding=1, stride=2)
 11            self.conv4 = nn.Conv2d(256, 512, kernel_size=3, padding=1, stride=2)
 12            self.conv5 = nn.Conv2d(512, 1024, kernel_size=3, padding=1, stride=2)
 13
 14            self.flatten = nn.Flatten(2)
 15
 16            # Add more layers...
 17            self.fc1 = nn.Linear(1024, 200) # 200 is the number of classes in TinyImageNet
 18
 19        def forward(self, x):
 20            x = self.conv1(x).relu()
 21            x = self.conv2(x).relu()
 22            x = self.conv3(x).relu()
 23            x = self.conv4(x).relu()
 24            x = self.conv5(x).relu()
 25
 26            x = self.flatten(x).mean(-1)
 27
 28            return self.fc1(x)
```

```
≡ requirements.txt U        🐍 train.py M ✕      🐍 customnet.py U

  🐍 train.py
❗ 1   from models.customnet import CustomNet
```

Step 7: Use add, commit and push to put your files back on github

```
● (robots) (base) davidebuoso@MacBook-Pro-di-Davide mldl_project_skeleton % git add .
● (robots) (base) davidebuoso@MacBook-Pro-di-Davide mldl_project_skeleton % git commit -m "first commit"
  [main d0de8e0] first commit
   3 files changed, 32 insertions(+)
   create mode 100644 models/customnet.py
   create mode 100644 requirements.txt
○ (robots) (base) davidebuoso@MacBook-Pro-di-Davide mldl_project_skeleton % git push
```

NOTE: Do not push any dataset on Github!

Once you finished to setup your project locally and pushed everything on Github you should be able to see your new files in the repository!

Now you can open the Colab, clone your repo, download the datasets (on colab) and run your code.

Your turn!

It's time to setup your project!

- At this link https://github.com/lambdavi/mldl_project_skeleton you can find the starting repo (TO FORK).
- Fork our repo and then clone your repo on your computer.
- Take your code from Lab1/Lab2 and populate the repo with Python files in the right directories.
- Add visualization logic.
- Test your code on Colab.

Notebook @
https://colab.research.google.com/drive/1sLHv8x3p_SOI
2Y6wo5sf1XaId329HxH4?usp=sharing