

致 亲爱的 Pytorch Learner：

这是 PyTorch 框架入门指南，主要以 ipynb 格式的教程为主，无论是学习还是运行、调试都很方便。本教程可以让你学会：pytorch 常见函数、模块的使用；深度学习框架的基本复现、小领域（nlp、cv 等）的实践、一些小项目的实践。祝学习愉快～

主要内容

本教程主要包含以下内容：



可以看出主要分为两部分：

基础部分（第 2~5 章）：介绍 PyTorch 基础语法和使用，这部份介绍了 PyTorch 中主要的模块，和深度学习中常用的一些工具。对于这部分内容，这里利用 Jupyter Notebook 作为教学工具，读者可以结合 notebook 修改运行，反复实验、调试。

- 第 2 章介绍 PyTorch 的安装以及相关学习环境的配置。同时，本章以概要的方式介绍 PyTorch 的主要内容，帮助读者初步了解 PyTorch。
- 第 3 章介绍 PyTorch 中多维数组 Tensor 以及自动微分系统 autograd 的使用，举例说明如何使用 Tensor 和 autograd 实现线性回归，并对比它们的不同点。本章对 Tensor 的基本结构以及 autograd 的原理进行了分析，帮助读者更加全面地了解 PyTorch 的底层模块。
- 第 4 章介绍 PyTorch 中神经网络模块 nn 的基本用法，讲解了神经网络中的层、激活函数、损失函数以及优化器等，在本章的最后带领读者使用不到 50 行代码实现经典的网络结构 ResNet。
- 第 5 章介绍 PyTorch 中的数据加载、预训练模型、可视化工具以及 GPU 加速等工具，合理地使用这些工具可以提高用户的编程效率。

高级扩展（第 6~8 章）：讲解 PyTorch 中的一些高级扩展，学习这部分内容可以帮助读者编写更加高效的程序。

- 第 6 章介绍 PyTorch 中的向量化思想，主要包括广播法则、基本索引、高级索引以及爱因斯坦操作。在本章的最后带领读者使用向量化思想实现深度学习中的卷积操作、交并比、RoI Align 以及反向 Unique 操作。
- 第 7 章介绍 PyTorch 中的分布式操作。分布式计算和并行计算可以加速网络的训练过程，本章详细介绍了并行计算和分布式计算的基本原理，同时介绍了如何使用 torch.distributed 以及 Horovod 进行 PyTorch 的分布式训练。
- 第 8 章介绍 PyTorch 中的 CUDA 扩展，带领读者使用 CUDA 实现 Sigmoid 函数。同时，本章对 CUDA、NVIDIA-driver、cuDNN 以及 Python 之间的关系进行了总结。

实战应用（第 9~13 章）：利用 PyTorch 实现了几个酷炫有趣的应用，对于这部分的内容，本仓库给出完整的实现代码，并提供预训练好的模型作为 demo，供读者测试。

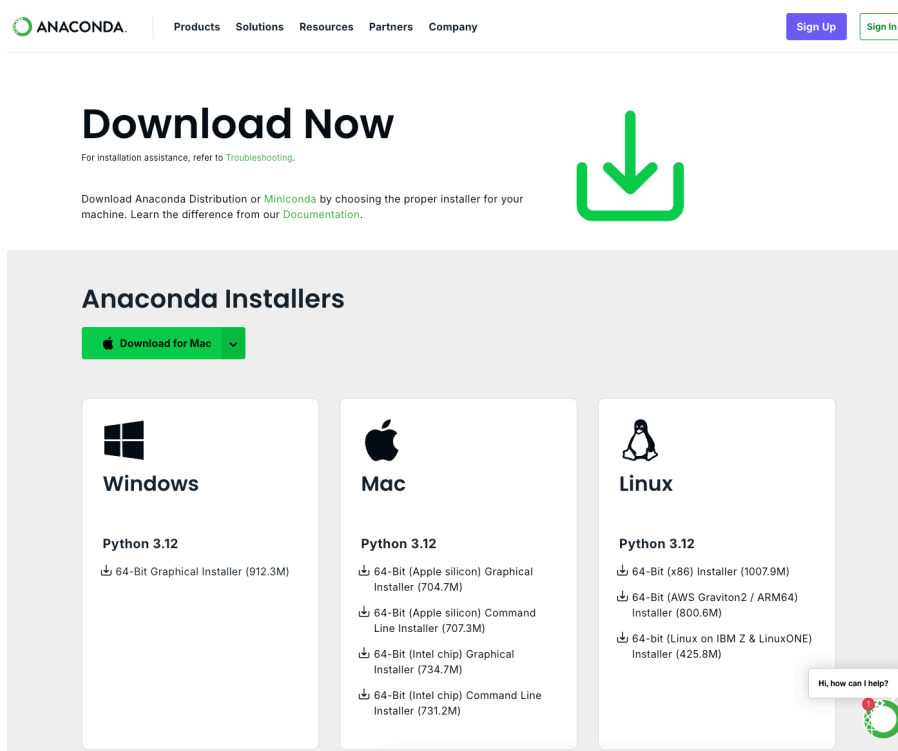
- 第 9 章是承上启下的一章，目标不是教会读者新函数、新知识，而实结合 Kaggle 中的一个经典比赛，实现深度学习中最为简单的图像二分类问题。在实现的过程中，将带领读者复习前 5 章的知识，并帮助读者合理地组织程序和代码，使程序更加易读且更好维护。同时，本章介绍了如何在 PyTorch 中调试。
- 第 10 章介绍生成对抗网络的基本原理，带领读者从零开始实现一个动漫头像生成器，能够利用生成对抗网络生成风格多变的动漫头像。
- 第 11 章介绍自然语言处理的一些基本知识，详细介绍了 CharRNN 以及 Transformer 的基本原理。本章带领读者使用 Transformer 实现自动写诗，该程序可以模仿古人进行诗词的续写以及藏头诗的生成。
- 第 12 章介绍风格迁移的基本原理，带领读者实现支持任意风格迁移的神经网络。通过该网络，读者可以将任意图片转换为名画的风格。
- 第 13 章介绍目标检测的基本原理，带领读者实现单阶段、无锚框、无非极大值抑制的

目标检测算法 CenterNet。CenterNet 的设计思路可以迁移到三维图像的目标检测、人体姿态估计以及目标追踪等经典的计算机视觉问题中。

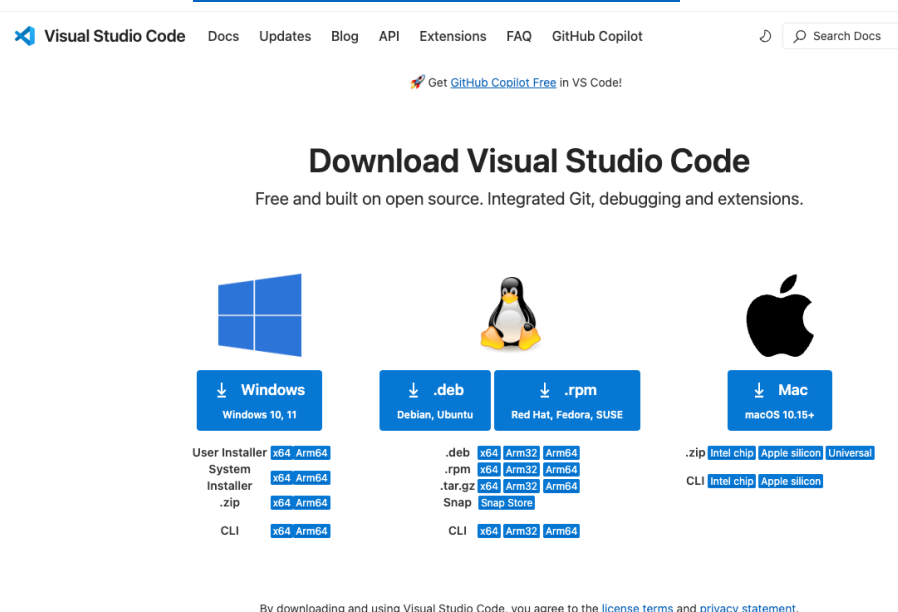
软件安装及环境配置

在正式使用这个教程之前, 我们需要先安装相应的软件 (Anaconda+Visual Studio Code), 然后进行环境的配置。

1. 下载并安装 Anaconda: <https://www.anaconda.com/download/success> , 根据自己的操作系统选择即可:

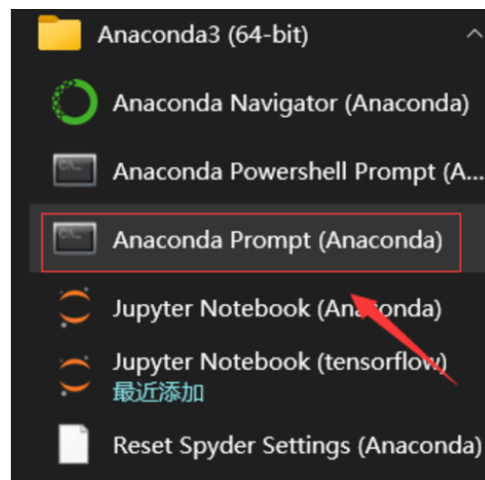


2. 下载并安装 VScode: <https://code.visualstudio.com/Download> :

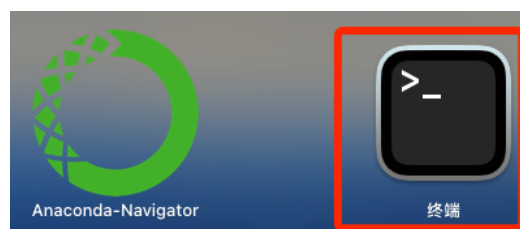


最新资料, 请认准淘宝店铺「牛博带你学 AI」

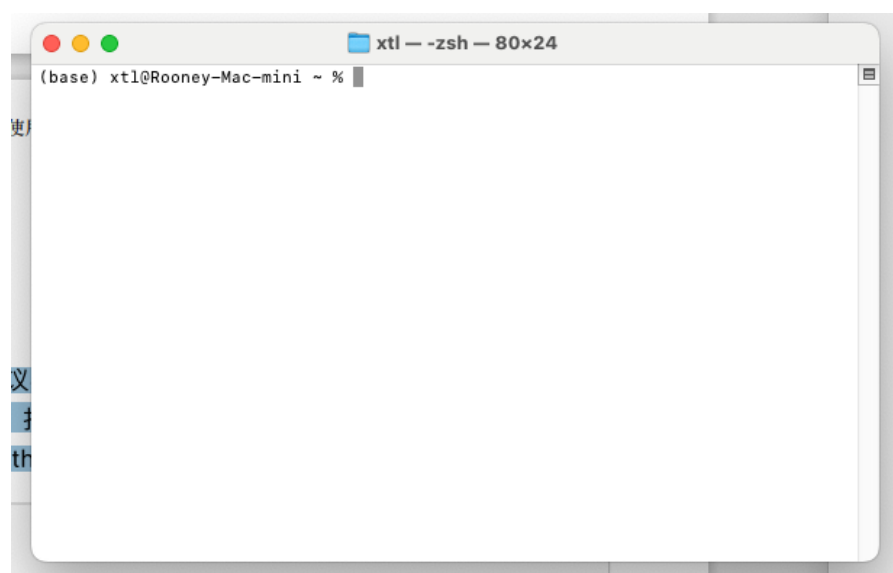
3. **创建环境**，Anaconda 安装完后，你会发现多了一些东西，例如 windows 系统：



Mac 系统：



接下来要创建环境，所有项目都依赖一个虚拟环境（不建议不同项目混用环境，容易出现不兼容的问题），我们使用conda 指令进行环境的搭建和配置。首先，打开 prompt (Mac 系统中是终端)，也就是图中红框中的东西，然后看到这样的程序：



然后输入以下指令：创建基于 Python=3.10 的环境，环境的名字叫做 env_1：

```
conda create -n env_q python=3.10
```

如下图所示，敲完后按回车：

```
xtl — python • conda create -n env_1 python=3.10 — 80x24
((base) xtl@Rooney-Mac-mini ~ % conda create -n env_1 python=3.10
Retrieving notices: ...working... done
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 23.7.4
  latest version: 25.1.1

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=25.1.1

## Package Plan ##

environment location: /Users/xtl/anaconda3/envs/env_1
```

按回车后
出现一堆文字

然后输入 y，并回车确认：

```
xtl — python • conda create -n env_1 python=3.10 — 80x24
xz-5.6.4 | h80987f9_1 289 KB
-----|-----
Total: 20.9 MB

The following NEW packages will be INSTALLED:

bzip2      pkgs/main/osx-arm64::bzip2-1.0.8-h80987f9_6
ca-certificates pkgs/main/osx-arm64::ca-certificates-2025.2.25-hca03da5_0
libffi     pkgs/main/osx-arm64::libffi-3.4.4-hca03da5_1
ncurses    pkgs/main/osx-arm64::ncurses-6.4-h313beb8_0
openssl    pkgs/main/osx-arm64::openssl-3.0.16-h02f6b3c_0
pip        pkgs/main/osx-arm64::pip-25.0-py310hca03da5_0
python     pkgs/main/osx-arm64::python-3.10.16-hb885b13_1
readline   pkgs/main/osx-arm64::readline-8.2-h1a28f6b_0
setuptools  pkgs/main/osx-arm64::setuptools-75.8.0-py310hca03da5_0
sqlite     pkgs/main/osx-arm64::sqlite-3.45.3-h80987f9_0
tk         pkgs/main/osx-arm64::tk-8.6.14-h6ba3021_0
tzdata     pkgs/main/noarch::tzdata-2025a-h04d1e81_0
wheel      pkgs/main/osx-arm64::wheel-0.45.1-py310hca03da5_0
xz         pkgs/main/osx-arm64::xz-5.6.4-h80987f9_1
zlib       pkgs/main/osx-arm64::zlib-1.2.13-h18a0788_1

Proceed ([y]/n)? y
```

输入 y，并按回车

确认后就开始自动安装相应的包了，安装完之后如下图：

```
xtl — -zsh — 80x24
tzdata      pkgs/main/noarch::tzdata-2025a-h04d1e81_0
wheel       pkgs/main/osx-arm64::wheel-0.45.1-py310hca03da5_0
xz          pkgs/main/osx-arm64::xz-5.6.4-h80987f9_1
zlib        pkgs/main/osx-arm64::zlib-1.2.13-h18a0788_1

Proceed ([y]/n)? y

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done

# To activate this environment, use
#
#   $ conda activate env_1
#
# To deactivate an active environment, use
#
#   $ conda deactivate

(base) xtl@Rooney-Mac-mini ~ %
```

成功安装！

4. 激活当前环境，从 base 环境切换到刚创建好的名为“env_1”的环境。使用如下指令：

```
conda activate env_1
```

然后就能看到环境从 base 环境切换到了新创建好的 env_1 环境：



```
xtl --zsh -- 80x24
wheel      pkgs/main/osx-arm64::wheel-0.45.1-py310hca03da5_0
xz         pkgs/main/osx-arm64::xz-5.6.4-h80987f9_1
zlib       pkgs/main/osx-arm64::zlib-1.2.13-h18a0788_1

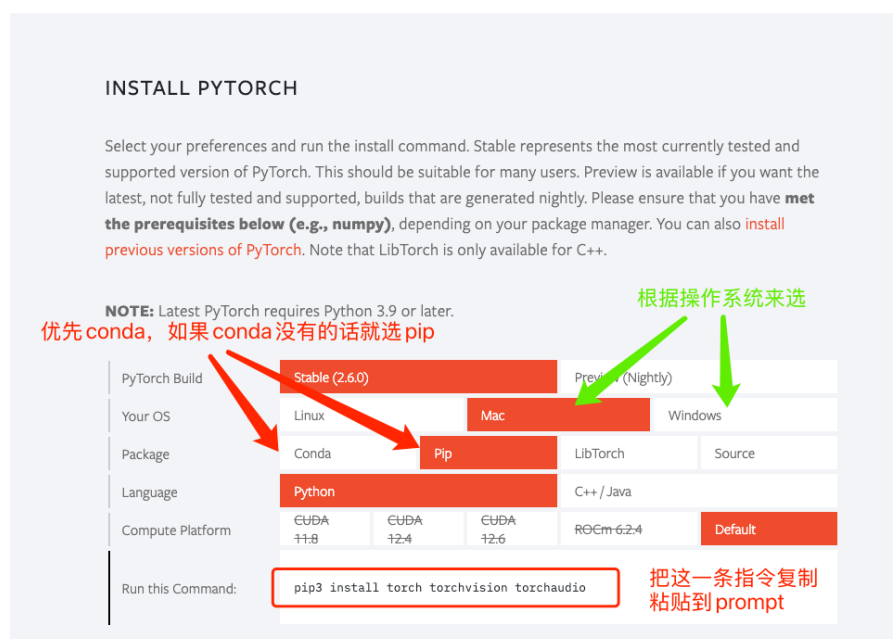
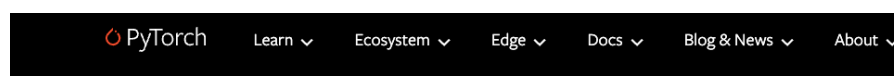
Proceed ([y]/n)? y

Downloading and Extracting Packages

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate env_1
#
# To deactivate an active environment, use
#
#     $ conda deactivate
#

[(base)] xtl@Rooney-Mac-mini ~ % conda activate env_1
[(env_1)] xtl@Rooney-Mac-mini ~ %
```

5. 在 env_1 环境中安装 Pytorch，首先进入 Pytorch 官网：<https://pytorch.org/>，下拉，并选择相应的安装方式：



INSTALL PYTORCH

Select your preferences and run the install command. Stable represents the most currently tested and supported version of PyTorch. This should be suitable for many users. Preview is available if you want the latest, not fully tested and supported, builds that are generated nightly. Please ensure that you have **met the prerequisites below (e.g., numpy)**, depending on your package manager. You can also **install previous versions of PyTorch**. Note that LibTorch is only available for C++.

NOTE: Latest PyTorch requires Python 3.9 or later.

优先 conda，如果 conda 没有的话就选 pip

根据操作系统来选

PyTorch Build	Stable (2.6.0)	Preview (Nightly)
Your OS	Linux	Mac
Package	Conda	Pip
Language	Python	C++ / Java
Compute Platform	CUDA 11.8	CUDA 12.4
Run this Command:	<code>pip3 install torch torchvision torchaudio</code>	

把这一条指令复制粘贴到 prompt

提醒： 牛博不推荐安装最新版本的 Pytorch，因为不稳定，运行常规项目很容易报错，所以，我们可以找到历史版本（<https://pytorch.org/get-started/previous-versions/>），往前退 5 个版本最佳，这里我们下拉并找到 v2.1.0 这个版本：

'''

```
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/main/
conda config --add channels https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkg/free/
conda config --set show_channel_urls yes
```

'''

在配置完清华镜像源后读者便可使用官网中去掉`-c pytorch`的命令较快地通过 conda 完成 PyTorch 的安装了。

有关 conda 的相关指令请收好，必有大用

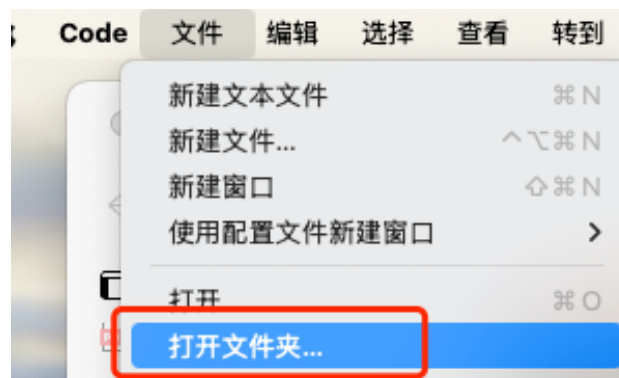
=====conda 的环境管理=====

conda info -e	查看当前已安装的环境
conda create -n name python=3.8	添加 3.8 版本的 Python 环境 位置在/envs/name
conda activate name	切换到 name 环境
conda deactivate	返回默认的 Python 环境
conda remove -n name --all	删除已安装的 Python 环境
conda create -n name2 --clone name1	克隆一个环境 name2

=====conda 的包管理=====

conda list	查看当前环境下已安装的包
conda list -n name	查看指定环境的已安装的包
conda search	某包查找某 package 信息
conda install -n name selenium	指定环境安装 package，不加-n 则安装在当前活跃环境
conda update -n name selenium	指定环境更新 package，不加-n 则更新在当前活跃环境
conda update --all	更新所有包
conda remove -n name selenium	删除 package，不加-n 则删除在当前活跃环境

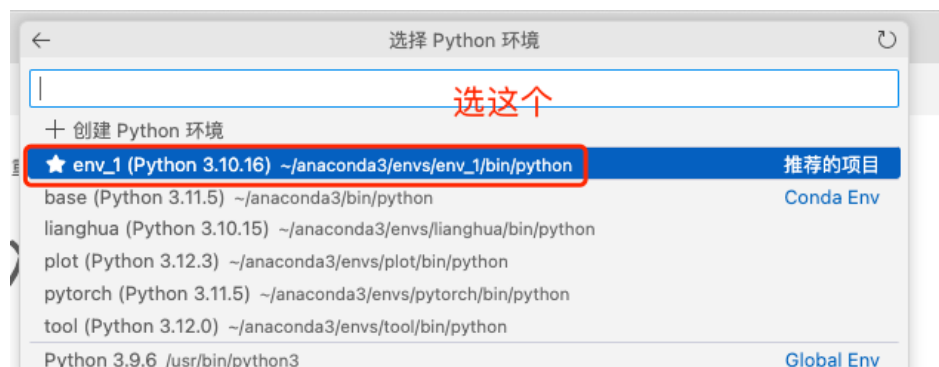
6. 打开 VScode，左上角，选择打开文件夹...，并选择【ipynb 教程这个文件夹】：



然后在左边打开相应的 ipynb 文件，开始学习：



记得先**选择内核**，这里的内核对应的是我们之前创建好的 **env_1** 环境：



本章将为读者提供一个快速入门PyTorch的指南，其中1.1节中笔者将带领读者快速浏览PyTorch中的主要内容，以便于读者建立一个关于PyTorch的大致印象。

那么接下来就可以开始快乐的跟着 ipynb 的教程开始学习了！对于每一个代码框，都建议反复的运行、调试，加深印象。学习过程中不懂的一些函数，也可以在百度-csdn 博客进行查阅，更精细地学习喔～

牛博已经把环境配置写的很详细了，如果你在照着这个这个教程还是出问题，并且百度也无法解决，可以来问牛博噢，在店铺客服处留言即可。