pytorch 及大杂烩等的基本了解实验报告

张宇鑫 22090032057

2024 年 9 月 15 日

1 实验目的

基本了解大杂烩（github,命令行，vpn 等），简单上手了解 pytorch 神经网络模型。

2 实验内容

大杂烩：

命令行参数：

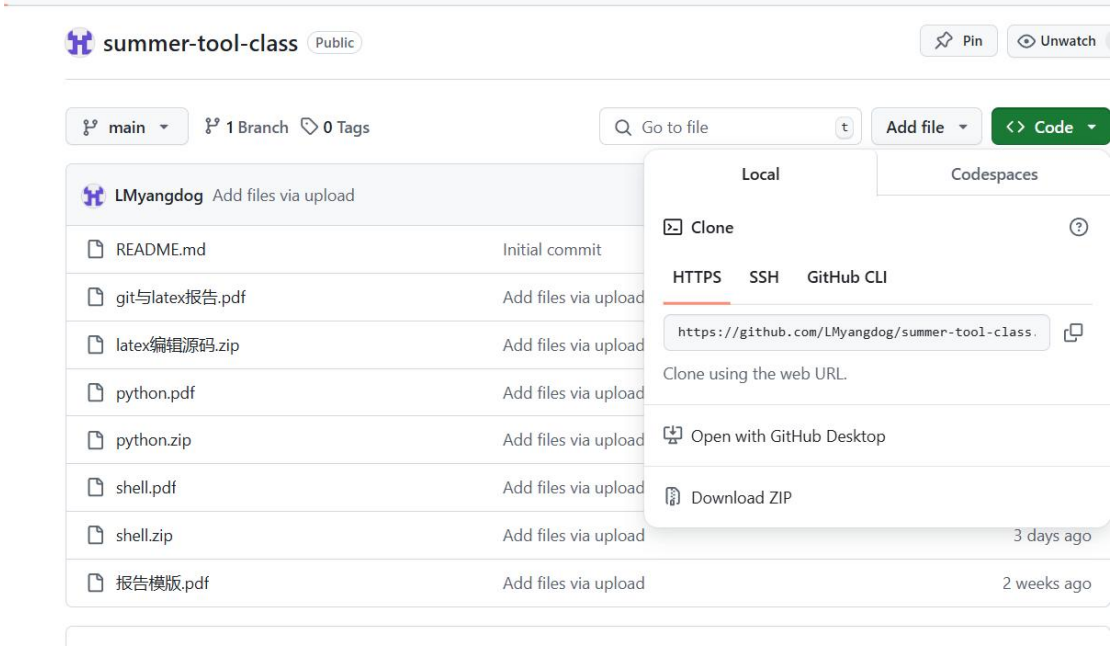
```
PS C:\Users\86157> python --version
Python 3.8.10
```

```
PS C:\Users\86157> pip --help

Usage:
  pip <command> [options]

Commands:
  install                     Install packages.
  download                    Download packages.
  uninstall                   Uninstall packages.
  freeze                      Output installed packages in requirements form
at.
  inspect                     Inspect the python environment.
  list                        List installed packages.
  show                        Show information about installed packages.
  check                       Verify installed packages have compatible depe
ndencies.
  config                      Manage local and global configuration.
```

Github:



Pytorch:

安装 pytorch 成功。

```
C:\Users\86157\PycharmProjects\pythonProject11\venv\Scripts\python.exe C:\Users\86157\PycharmProjects\pythonProject8\python3.py
tensor([[0.7522, 0.8729, 0.3935],
        [0.8035, 0.3636, 0.9156],
        [0.3339, 0.3976, 0.9883],
        [0.1326, 0.9834, 0.3113],
        [0.0456, 0.1025, 0.4598]])
```

运行实例：

导入所需 torch

```
import torch
import torch.nn as nn
import torchvision
import torch.utils.data as Data
```

CNN 编写：

```python
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()

        self.conv1 = nn.Sequential(
            nn.Conv2d(  # 输入的图片 (1, 28, 28)
                in_channels=1,
                out_channels=16,  # 经过一个卷积层之后 (16,28,28)
                kernel_size=5,
                stride=1,  # 如果想要 con2d 出来的图片长宽没有变化, padding=(kernel_size-1)/2 当 stride=1
                padding=2
            ),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2)  # 经过池化层处理，维度为（16,14,14）
        )

        self.conv2 = nn.Sequential(
            nn.Conv2d(  # 输入（16,14,14）
                in_channels=16,
                out_channels=32,
                kernel_size=5,
                stride=1,
                padding=2
            ),  # 输出（32,14,14）
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2)  # 输出（32,7,7）
        )

```

打印结果查看是否运行成功：

```python
# 打印前十个测试结果和真实结果进行对比
test_output = cnn(test_x[:10])
pred_y = torch.max(test_output, 1)[1].numpy()
print(pred_y, 'prediction number')
print(test_y[:10].numpy(), 'real number')
```

查看运行结果：运行成功，配置环境没问题

```
Epoch:  0 | train loss: 0.0302
Epoch:  0 | train loss: 0.0197
Epoch:  0 | train loss: 0.0729
[7 2 1 0 4 1 4 9 5 9] prediction number
[7 2 1 0 4 1 4 9 5 9] real number
```

3 实验结果

Pytorch 安装成功，样例程序成功运行

```
Epoch:  0 | train loss: 0.0302
Epoch:  0 | train loss: 0.0197
Epoch:  0 | train loss: 0.0729
[7 2 1 0 4 1 4 9 5 9] prediction number
[7 2 1 0 4 1 4 9 5 9] real number
```

4 实验感悟

通过本次实验，我对于很多计算机的基本工具与技巧有了更加深入的了解，同时也学会了更多的基本工具的使用，同时了解到了神经网络的相关内容，学会了基本的训练模型构建，并且成功配置好了所需的环境，总之这次实验学到了很多东西。