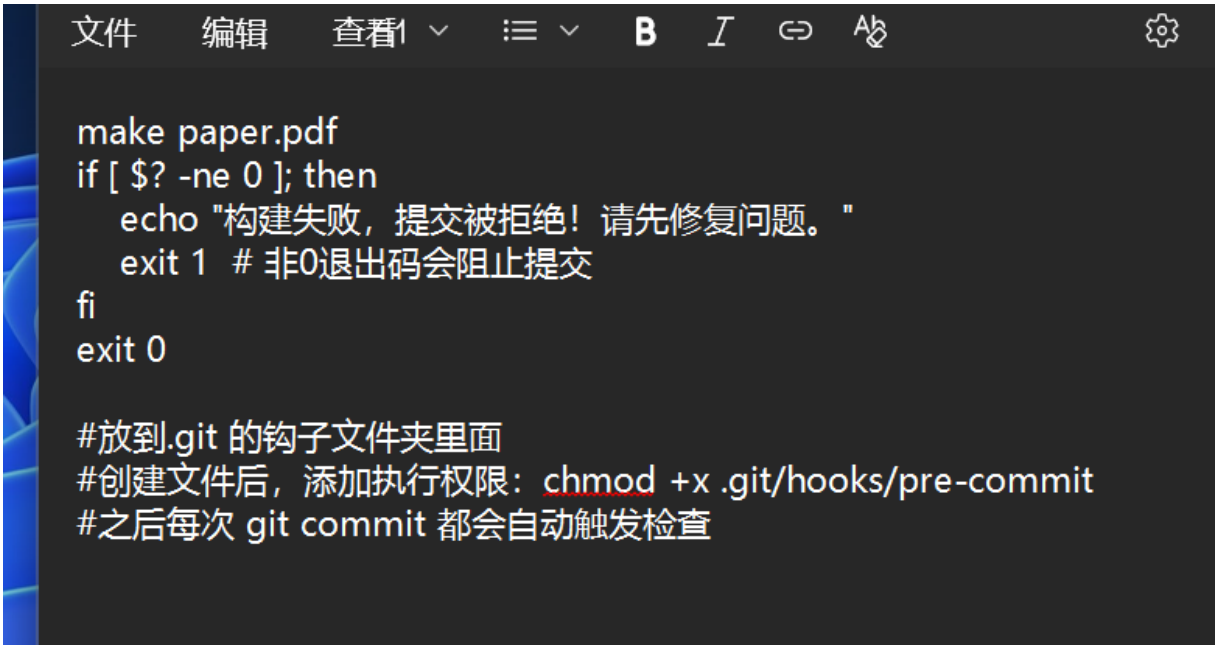
系统开发工具第四次实验报告

24020007044 侯余铂

2025 年 9 月 23 日

以下是 20 个案例:

1. **将该文件放入到.git 文件的钩子文件中，之后每次提交都会自动检查提交是否是正确**



2. **将该文件放入到项目的根目录下面去，直接可以通过 make 来构建论文**

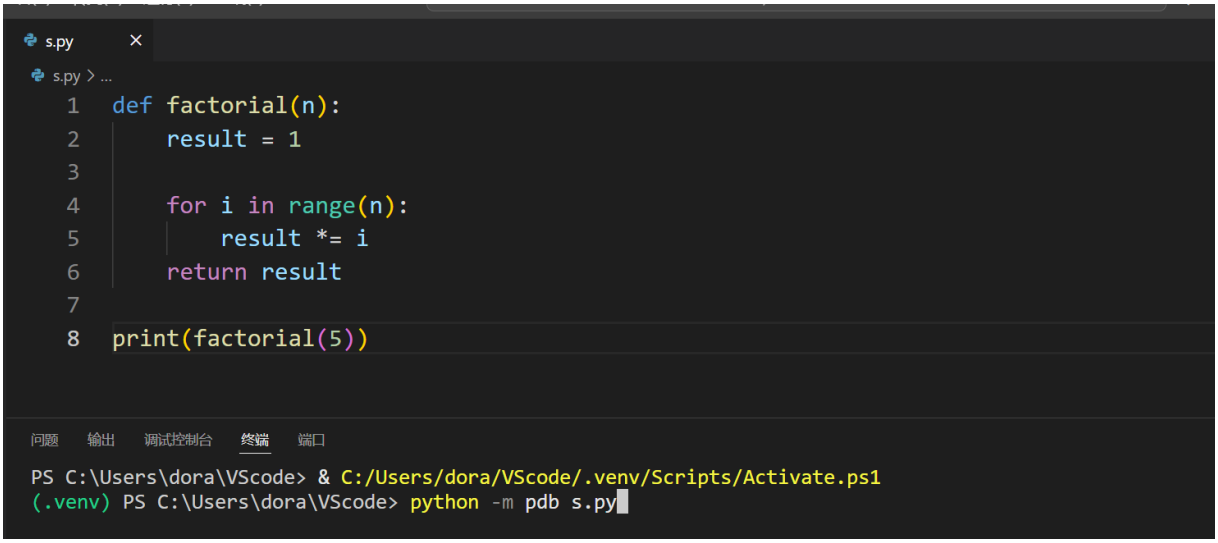
```
paper.pdf: paper.tex plot-data.png
	pdflatex paper.tex

plot-%.png: %.dat plot.py
	./plot.py -i $*.dat -o $@

.PHONY: clean
clean:
	rm -f paper.pdf plot-*.png


#放到项目根目录下
#构建论文：make
#清理文件：make clean
```

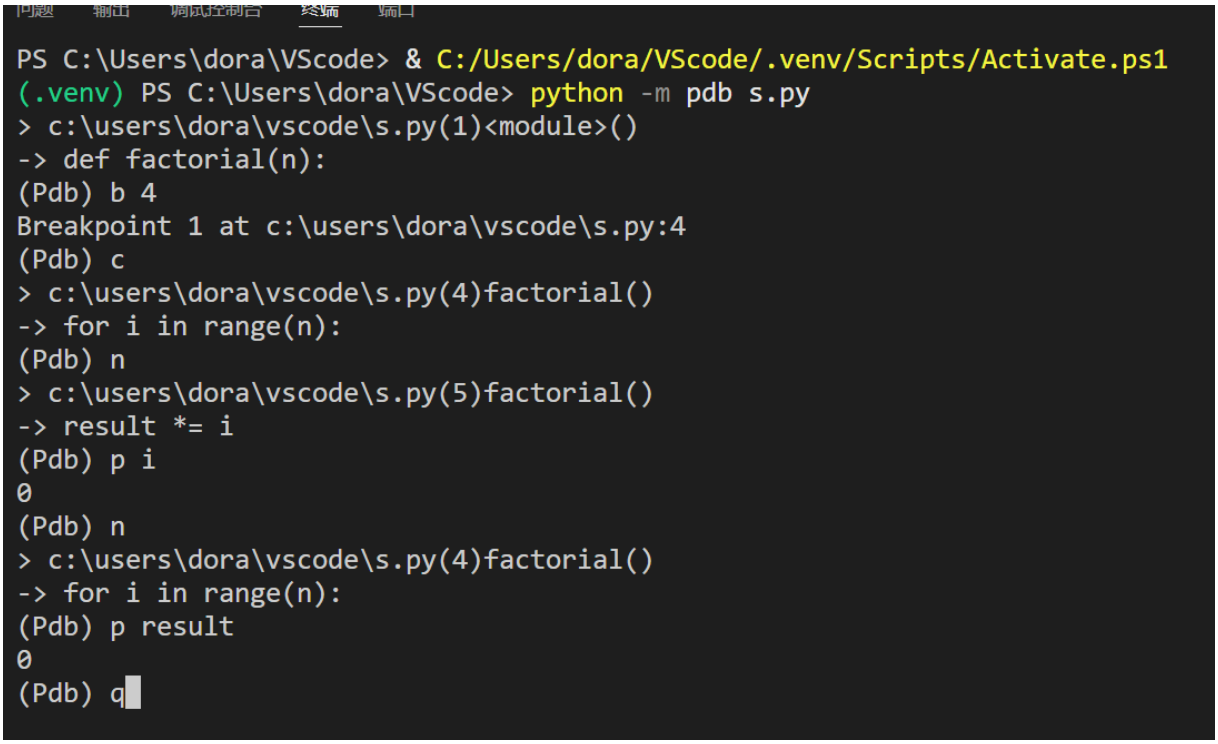
3. 用日志查看通过使用 sudo 指令的使用情况

```
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
dorasweater@ubuntu:~$ journalctl | grep sudo
```

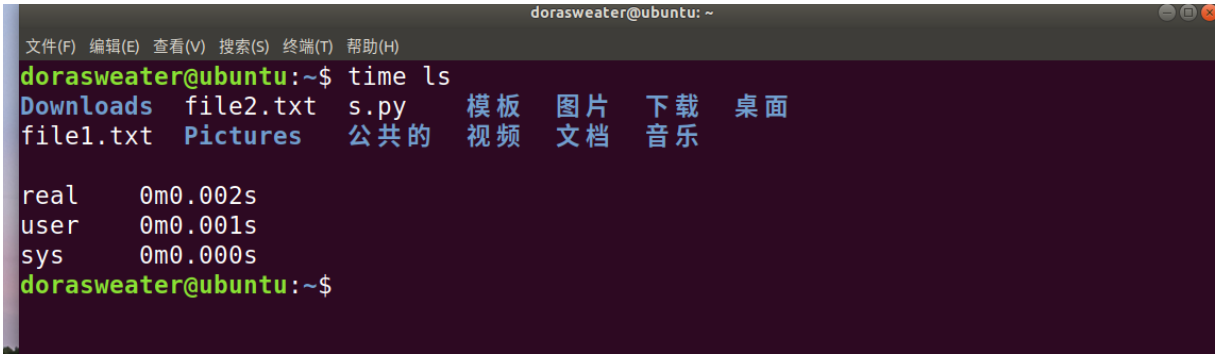
```
er root by (uid=0)
9月  14 07:36:33 ubuntu sudo[2920]: pam_unix(sudo:session): session closed for us
er root
9月  14 07:36:35 ubuntu sudo[2922]: dorasweater : TTY=pts/0 ; PWD=/home/dorasweat
er ; USER=root ; COMMAND=/bin/rm /var/lib/dpkg/lock
9月  14 07:36:35 ubuntu sudo[2922]: pam_unix(sudo:session): session opened for us
er root by (uid=0)
9月  14 07:36:35 ubuntu sudo[2922]: pam_unix(sudo:session): session closed for us
er root
9月  14 07:37:03 ubuntu sudo[2924]: dorasweater : TTY=pts/0 ; PWD=/home/dorasweat
er ; USER=root ; COMMAND=/usr/bin/apt-get install tmux
9月  14 07:37:03 ubuntu sudo[2924]: pam_unix(sudo:session): session opened for us
er root by (uid=0)
9月  14 07:37:21 ubuntu sudo[2924]: pam_unix(sudo:session): session closed for us
er root
9月  14 07:37:53 ubuntu sudo[3636]: dorasweater : TTY=pts/0 ; PWD=/home/dorasweat
er ; USER=root ; COMMAND=/usr/bin/apt-get install tmuxbrew install tmux
9月  14 07:37:53 ubuntu sudo[3636]: pam_unix(sudo:session): session opened for us
er root by (uid=0)
9月  14 07:37:54 ubuntu sudo[3636]: pam_unix(sudo:session): session closed for us
er root
dorasweater@ubuntu:~$
```

4. 通过 python 的 pdb 来进行对 python 文件的调试





5. 在 linux 系统中通过 time 来进行对文件的时间测试，来判断性能的优劣



```
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
dorasweater@ubuntu:~$ time ls
Downloads  file2.txt  s.py    模板  图片  下载  桌面
file1.txt  Pictures   公共的  视频  文档  音乐

real    0m0.002s
user    0m0.002s
sys     0m0.000s
dorasweater@ubuntu:~$ time python3 s.py
120

real    0m0.030s
user    0m0.026s
sys     0m0.004s
dorasweater@ubuntu:~$
```

6. 用 ping 来查看网站的联通情况，是否是正常运行的

```
文件(F)  编辑(E)  查看(V)  搜索(S)  终端(T)  帮助(H)
dorasweater@ubuntu:~$ ping www.baidu.com
PING www.baidu.com (180.101.49.44) 56(84) bytes of data.
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=1 ttl=128 time=24.0 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=2 ttl=128 time=32.7 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=3 ttl=128 time=22.8 ms
S64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=4 ttl=128 time=24.3 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=5 ttl=128 time=22.1 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=6 ttl=128 time=35.0 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=7 ttl=128 time=62.7 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=8 ttl=128 time=23.4 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=9 ttl=128 time=25.1 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=10 ttl=128 time=32.4 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=11 ttl=128 time=25.1 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=12 ttl=128 time=23.3 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=13 ttl=128 time=23.2 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=14 ttl=128 time=27.7 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=15 ttl=128 time=22.6 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=16 ttl=128 time=24.8 ms
64 bytes from 180.101.49.44 (180.101.49.44): icmp_seq=17 ttl=128 time=24.4 ms
^C
--- www.baidu.com ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16038ms
rtt min/avg/max/mdev = 22.107/28.029/62.702/9.446 ms
dorasweater@ubuntu:~$
```

7. 通过终端来下载 pytorch 到本地，通过指令就行

```
问题  输出  调试控制台  终端  端口
PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\dora\VScode> pip install torch torchvision torchaudio
```

```
Downloading torch-2.8.0-cp312-cp312-win_amd64.whl (241.3 MB)
   ---------------------------------------- 241.3/241.3 MB 4.8 MB/s  0:00:50
Downloading torchvision-0.23.0-cp312-cp312-win_amd64.whl (1.6 MB)
   ---------------------------------------- 1.6/1.6 MB 4.1 MB/s  0:00:00
Downloading torchaudio-2.8.0-cp312-cp312-win_amd64.whl (2.5 MB)
   ---------------------------------------- 2.5/2.5 MB 5.5 MB/s  0:00:00
Downloading sympy-1.14.0-py3-none-any.whl (6.3 MB)
   ---------------------------------------- 6.3/6.3 MB 5.1 MB/s  0:00:01
Downloading mpmath-1.3.0-py3-none-any.whl (536 kB)
   ---------------------------------------- 536.2/536.2 kB 5.8 MB/s  0:00:00
Downloading typing_extensions-4.15.0-py3-none-any.whl (44 kB)
Downloading filelock-3.19.1-py3-none-any.whl (15 kB)
Downloading fsspec-2025.9.0-py3-none-any.whl (199 kB)
Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)
Downloading MarkupSafe-3.0.2-cp312-cp312-win_amd64.whl (15 kB)
Using cached setuptools-80.9.0-py3-none-any.whl (1.2 MB)
Installing collected packages: mpmath, typing-extensions, sympy, setuptools, MarkupSafe, fsspec, filelock, jinja2, torch, torchvision, torchaud
io
Successfully installed MarkupSafe-3.0.2 filelock-3.19.1 fsspec-2025.9.0 jinja2-3.1.6 mpmath-1.3.0 setuptools-80.9.0 sympy-1.14.0 torch-2.8.0 to
rchaudio-2.8.0 torchvision-0.23.0 typing-extensions-4.15.0
(.venv) PS C:\Users\dora\VScode>
```

8. 为 pythorch 配备上运行的 GPU

```
PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\dora\VScode> pip install torch torchvision torchaudio
```

```
PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\dora\VScode> pip install torch torchvision torchaudio
Requirement already satisfied: torch in c:\users\dora\vscode\.venv\lib\site-packages (2.8.0)
Requirement already satisfied: torchvision in c:\users\dora\vscode\.venv\lib\site-packages (0.23.0)
Requirement already satisfied: torchaudio in c:\users\dora\vscode\.venv\lib\site-packages (2.8.0)
Requirement already satisfied: filelock in c:\users\dora\vscode\.venv\lib\site-packages (from torch) (3.19.1)
Requirement already satisfied: typing-extensions>=4.10.0 in c:\users\dora\vscode\.venv\lib\site-packages (from torch) (4.15.0)
Requirement already satisfied: sympy>=1.13.3 in c:\users\dora\vscode\.venv\lib\site-packages (from torch) (1.14.0)
Requirement already satisfied: networkx in c:\users\dora\vscode\.venv\lib\site-packages (from torch) (3.5)
Requirement already satisfied: jinja2 in c:\users\dora\vscode\.venv\lib\site-packages (from torch) (3.1.6)
Requirement already satisfied: fsspec in c:\users\dora\vscode\.venv\lib\site-packages (from torch) (2025.9.0)
Requirement already satisfied: setuptools in c:\users\dora\vscode\.venv\lib\site-packages (from torch) (80.9.0)
Requirement already satisfied: numpy in c:\users\dora\vscode\.venv\lib\site-packages (from torchvision) (2.2.6)
Requirement already satisfied: pillow!=8.3.*,>=5.3.0 in c:\users\dora\vscode\.venv\lib\site-packages (from torchvision) (11.3.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in c:\users\dora\vscode\.venv\lib\site-packages (from sympy>=1.13.3->torch) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\dora\vscode\.venv\lib\site-packages (from jinja2->torch) (3.0.2)
(.venv) PS C:\Users\dora\VScode>
```

9. 这是对系统里面是否下载好 pytorch 版本的检测

s.py

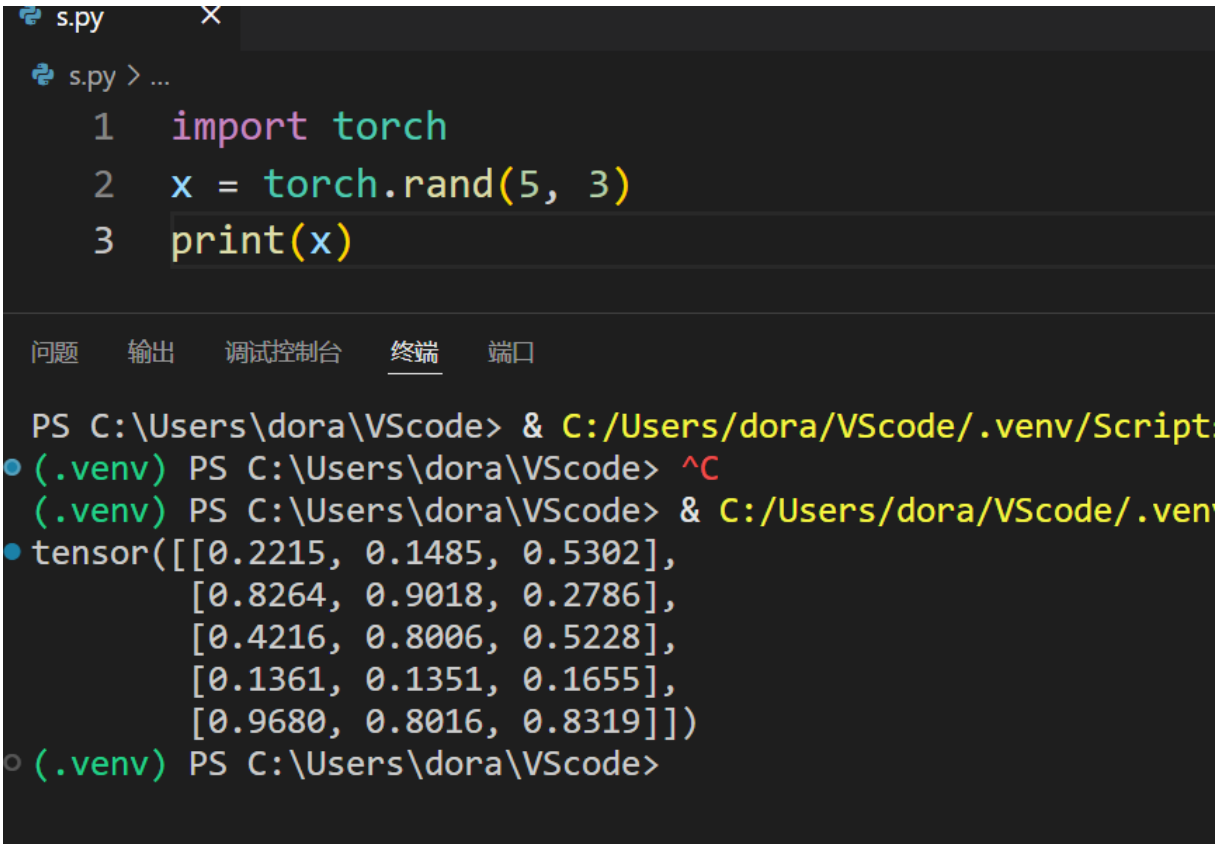
```
1  import torch
2
3  # 当前安装的 PyTorch 库的版本
4  print(torch.__version__)
5
```

问题 输出 调试控制台 终端 端口

```
PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scrip
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.ve
2.8.0+cpu
(.venv) PS C:\Users\dora\VScode>
```

10. 这是 pytorch 中将 numpy 向量转换成张量的过程

11. 这是 pytorch 中输出一个随机张量的过程



12. 这是 pytorch 中关于张量的加法和元素乘法的运算

```python
# 张量相加
import torch

e = torch.randn(2, 3)
f = torch.randn(2, 3)
print(e + f)

# 逐元素乘法
print(e * f)
```

```
问题  输出  调试控制台  终端  端口
(.venv) PS C:\Users\dora\VScode>

PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\dora\VScode> ^C
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.exe c:/Users/dora/VScode/s.py
(.venv) PS C:\Users\dora\VScode> ^C
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.exe c:/Users/dora/VScode/s.py
tensor([[ 0.5168,  0.0947, -0.0108],
        [-0.1779,  1.9183, -0.7525]])
tensor([[ 0.0229, -0.0275, -0.0261],
        [-1.3672,  0.9200, -1.8141]])
(.venv) PS C:\Users\dora\VScode>
```

13. pytorch 中定义一个简单的全连接的神经网络模型

```python
import torch
import torch.nn as nn
# 定义一个简单的全连接神经网络模型
class SimpleNN(nn.Module):
    def __init__(self):
        super(SimpleNN, self).__init__()
        # 定义一个输入层到隐藏层的全连接层
        self.fc1 = nn.Linear(2, 2)  # 输入 2 个特征，输出 2 个特征
        # 定义一个隐藏层到输出层的全连接层
        self.fc2 = nn.Linear(2, 1)  # 输入 2 个特征，输出 1 个预测值

    def forward(self, x):
        # 前向传播过程
        x = torch.relu(self.fc1(x))  # 使用 ReLU 激活函数
        x = self.fc2(x)  # 输出层
        return x
# 创建模型实例
model = SimpleNN()
# 打印模型
print(model)
```

```
问题  输出  调试控制台  终端  端口
tensor([[ 0.0229, -0.0275, -0.0261],
(.venv) PS C:\Users\dora\VScode> ^C
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.exe c:/Users/dora/VScode/s.py
SimpleNN(
  (fc1): Linear(in_features=2, out_features=2, bias=True)
  (fc2): Linear(in_features=2, out_features=1, bias=True)
)
(.venv) PS C:\Users\dora\VScode>
```

14. pytorch 中对张量的索引和获取子张量

s.py > ...

```python
import torch

x = torch.tensor([[10, 20, 30],
                  [40, 50, 60],
                  [70, 80, 90]])
row = x[1]
col = x[:, 2]
sub_tensor = x[0:2, 1:3]
print("第 2 行:\n", row)
print("第 3 列:\n", col)
print("子张量:\n", sub_tensor)
```

s.py > ...

```python
import torch

x = torch.tensor([[10, 20, 30],
                  [40, 50, 60],
                  [70, 80, 90]])
row = x[1]
col = x[:, 2]
sub_tensor = x[0:2, 1:3]
print("第 2 行:\n", row)
print("第 3 列:\n", col)
print("子张量:\n", sub_tensor)
```

问题 输出 调试控制台 终端 端口

```
(.venv) PS C:\Users\dora\VScode> ^C
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.exe c:/Users/dora/VScode/s.py
第 2 行:
 tensor([40, 50, 60])
第 3 列:
 tensor([30, 60, 90])
子张量:
 tensor([[20, 30],
        [50, 60]])
(.venv) PS C:\Users\dora\VScode>
```

15. pytorch 中对张量的重塑的过程

```python
import torch
x = torch.arange(1, 7)
y = x.view(2, 3)
z = x.view(-1, 2)
print("原始张量 x:\n", x)
print("重塑为 2x3:\n", y)
print("自动计算行数，列数为 2:\n", z)
```

```
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.ex
原始张量 x:
 tensor([1, 2, 3, 4, 5, 6])
重塑为 2x3:
 tensor([[1, 2, 3],
        [4, 5, 6]])
自动计算行数，列数为 2:
 tensor([[1, 2],
        [3, 4],
        [5, 6]])
(.venv) PS C:\Users\dora\VScode>
```

16. 在 pytorch 中生成一个线性相关的数据集

```python
import torch
import numpy as np
import matplotlib.pyplot as plt
# 随机种子，确保每次运行结果一致
torch.manual_seed(42)
# 生成训练数据
X = torch.randn(100, 2)
true_w = torch.tensor([2.0, 3.0])
true_b = 4.0
Y = X @ true_w + true_b + torch.randn(100) * 0.1
print(X[:5])
print(Y[:5])
```

```
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.exe c:/Use
tensor([[ 1.9269,  1.4873],
        [ 0.9007, -2.1055],
        [ 0.6784, -1.2345],
        [-0.0431, -1.6047],
        [-0.7521,  1.6487]])
tensor([12.4460, -0.4663,  1.7666, -0.9357,  7.4781])
(.venv) PS C:\Users\dora\VScode>
```

17. 在 pytorch 中生成一个线性回归的模型

```python
import torch.nn as nn

# 定义线性回归模型
class LinearRegressionModel(nn.Module):
    def __init__(self):
        super(LinearRegressionModel, self).__init__()
        self.linear = nn.Linear(2, 1)
    def forward(self, x):
        return self.linear(x)
model = LinearRegressionModel()
```

```
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.exe c:/Users/dora/VScode/s.py
(.venv) PS C:\Users\dora\VScode>
```

18. 在 pytorch 中对图像数据集的应用转换

```python
from torchvision import datasets, transforms
from torch.utils.data import DataLoader
transform = transforms.Compose([
    transforms.Resize((128, 128)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.5], std=[0.5])
])
train_dataset = datasets.MNIST(root='./data', train=True, transform=transform, download=True)
train_loader = DataLoader(dataset=train_dataset, batch_size=32, shuffle=True)
for images, labels in train_loader:
    print("图像张量大小:", images.size())
    break
```

```
PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/Activate.ps1
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.exe c:/Users/dora/VScode/t.py
100.0%
100.0%
100.0%
100.0%
图像张量大小: torch.Size([32, 1, 128, 128])
(.venv) PS C:\Users\dora\VScode>
```

19. 在 pytorch 中通过 dateset 来自定义数据集

```python
import torch
from torch.utils.data import Dataset
class MyDataset(Dataset):
    def __init__(self, data, labels):
        self.data = data
        self.labels = labels
    def __len__(self):
        return len(self.data)
    def __getitem__(self, idx):
        sample = self.data[idx]
        label = self.labels[idx]
        return sample, label
data = torch.randn(100, 5)
labels = torch.randint(0, 2, (100,))
dataset = MyDataset(data, labels)
print("数据集大小:", len(dataset))
print("第 0 个样本:", dataset[0])
```

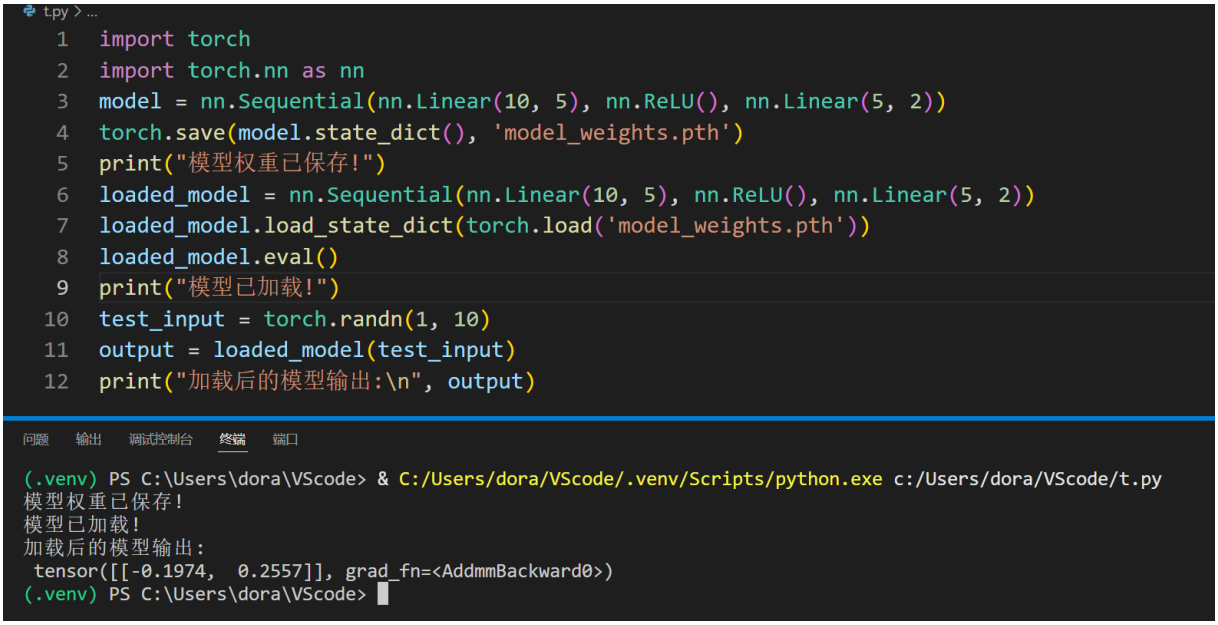
```
(.venv) PS C:\Users\dora\VScode> ^C
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.exe c:/Users/dora/VScode/t.py
数据集大小: 100
第 0 个样本: (tensor([-1.2254, -0.4263,  1.4276,  0.4771, -0.2233]), tensor(0))
(.venv) PS C:\Users\dora\VScode>
```

20. 在 pytorch 中自动求导的功能的基本实现

```python
import torch
x = torch.randn(2, 2, requires_grad=True)
print(x)
y = x + 2
z = y * y * 3
out = z.mean()
print(out)
```

```
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts
tensor([[ 1.0839, -1.6804],
        [-1.9987, -0.0793]], requires_grad=True)
tensor(9.9762, grad_fn=<MeanBackward0>)
(.venv) PS C:\Users\dora\VScode>
```

21. 在 pytorch 中对模型的基本保存和加载



```python
import torch
import torch.nn as nn
model = nn.Sequential(nn.Linear(10, 5), nn.ReLU(), nn.Linear(5, 2))
torch.save(model.state_dict(), 'model_weights.pth')
print("模型权重已保存!")
loaded_model = nn.Sequential(nn.Linear(10, 5), nn.ReLU(), nn.Linear(5, 2))
loaded_model.load_state_dict(torch.load('model_weights.pth'))
loaded_model.eval()
print("模型已加载!")
test_input = torch.randn(1, 10)
output = loaded_model(test_input)
print("加载后的模型输出:\n", output)
```

```
(.venv) PS C:\Users\dora\VScode> & C:/Users/dora/VScode/.venv/Scripts/python.exe c:/Users/dora/VScode/t.py
模型权重已保存!
模型已加载!
加载后的模型输出:
 tensor([[-0.1974,  0.2557]], grad_fn=<AddmmBackward0>)
(.venv) PS C:\Users\dora\VScode>
```

通过本次的学习，让我学会了关于元编程的很多内容，同时也学会了一些关于调试和分析的方法，对于不同的情况使用什么方法会更有效一些，同时还学习了解了一些关于 pytorch 的很多内容，对于张量的输出和相关的运算，以及张量的重置，还有如何搭建神经网络，和如何建立数据集及其处理该数据集的模型等等，通过本次的学习，让我学会了很多的知识

GitHub:git@github.com:Dora-pt/git_study.git