1. **WSL to Ubuntu**

1.1 run powershell as administrator enable WSL: wsl --install

1.2 Install Ubuntu: wsl --install -d Ubuntu-22.04

1.3 Checking: wsl --list –verbose

1. **Move the Virtual Machine from C drive to E drive**

2.1 Creating the folder: mkdir E:\WSL\Ubuntu-24.04

2.2 Exporting compressed virtual machine: wsl --export Ubuntu-24.04 E:\WSL\ubuntu-24.04-backup.tar

2.3 Unregistering: wsl --unregister Ubuntu-24.04 (for verification: wsl --list)

2.4 Import to E drive: wsl --import Ubuntu-24.04 E:\WSL\Ubuntu-24.04 E:\WSL\ubuntu-24.04-backup.tar --version 2

2.5 Activating: wsl -d Ubuntu-24.04

2.6 Cleanup: del E:\WSL\ubuntu-24.04-backup.tar

1. **Python 3.10 installation for virtual environment**
   1. Fucking with the Fire Wall of the Great China
      1. Backup of source: sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
      2. Editing the source: sudo nano /etc/apt/sources.list
      3. Replacing the content in the source:

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ noble main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ noble-updates main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ noble-backports main restricted universe multiverse

deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ noble-security main restricted universe multiverse

* 1. Downloading from Huawei mirror: curl -I <https://mirrors.tuna.tsinghua.edu.cn/python/3.10.12/Python-3.10.12.tar.xz>
  2. Downloading dependencies: sudo apt update

sudo apt install -y build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev curl libbz2-dev

* 1. Extracting the python file: tar -xf Python-3.10.12.tar.xz
  2. Compiling:

cd Python-3.10.12./configure --enable-optimizations make -j$(nproc)

sudo make install

3.6 Installing pip for python 3.10: python3.10 get-pip.py -i https://pypi.tuna.tsinghua.edu.cn/simple

1. **Virtual Environment:**

cd /home/acky/S-D-Mamba-main

source venv/bin/activate

1. **Requirement Installation**
   1. pip install torch==2.1.0+cu118 torchvision==0.16.0+cu118 torchaudio==2.1.0+cu118 -f https://download.pytorch.org/whl/torch\_stable.html
   2. pip install transformers==4.35.0
   3. For verification:

import torch

print("Torch:", torch.\_\_version\_\_)

print("CUDA:", torch.cuda.is\_available())

import mamba\_ssm

print("Mamba SSM successfully imported.")

* 1. Modify the requirement:

scikit-learn==1.3.0

numpy==1.26.4

matplotlib==3.7.0

torch==2.1.0+cu118

torchvision==0.16.0+cu118

torchaudio==2.1.0+cu118

reformer-pytorch==1.4.4

mamba-ssm==1.2.0

transformers==4.35.0

* 1. pip install --no-deps --force-reinstall -r requirements.txt

1. **Open the project in VS code**

6.1 In VS code terminal: wsl

6.2 In VS code terminal: cd /home/acky/S-D-Mamba-main

6.3 In VS code terminal: code .

6.4 In order to get the correct and synchronized connection to WSL

Go to Command Palette (Ctrl+Shift+P)

Manually add or delete the running project

6.5 Run WSL: Close Remote Connection

6.6 Remember to type Ctrl + S after editing the file

1. **Run the dummy dataset**

python3 run.py \

--is\_training 1 \

--model\_id bp\_s\_mamba\_exp1 \

--model S\_Mamba \

--data bp \

--root\_path ./data/ \

--features S \

--target SBP \

--seq\_len 128 \

--label\_len 64 \

--pred\_len 64 \

--use\_gpu True

1. **Run the real dataset**

This is for calling MAMBA

python3 run.py \

--is\_training 1 \

--model\_id bp\_mamba\_sdbp \

--model S\_Mamba \

--data bp \

--root\_path ./data/ \

--features S \

--target SBP \

--seq\_len 1000 \

--label\_len 64 \

--pred\_len 1 \

--enc\_in 1 \

--dec\_in 1 \

--c\_out 2 \

--use\_gpu True

This is for calling Transformer

python3 run.py \

--is\_training 1 \

--model\_id bp\_transformer\_sdbp \

--model Transformer \

--data bp \

--root\_path ./data/ \

--features S \

--target SBP \

--seq\_len 1000 \

--label\_len 64 \

--pred\_len 1 \

--enc\_in 1 \

--dec\_in 1 \

--c\_out 2 \

--use\_gpu True

1. **Resume training**

python3 run.py \

--is\_training 1 \

--model\_id bp\_mamba\_sdbp \

--model S\_Mamba \

--data bp \

--root\_path ./data/ \

--features S \

--target SBP \

--seq\_len 1000 \

--label\_len 64 \

--enc\_in 1 \

--dec\_in 1 \

--c\_out 2 \

--use\_gpu True \

--resume

1. **One-shot way to resume training with different dataset**

python3 extract\_real\_data\_to\_csv.py

python3 check\_and\_remove.py

python3 run.py \

--is\_training 1 \

--model\_id bp\_mamba\_sdbp \

--model S\_Mamba \

--data bp \

--root\_path ./data/ \

--features S \

--target SBP \

--seq\_len 1000 \

--label\_len 64 \

--enc\_in 1 \

--dec\_in 1 \

--c\_out 2 \

--use\_gpu True \

1. **Upload folders (entire project) to GitHub**

cd "E:\New folder1\S-D-Mamba-main"

# Initialize Git (only if not already a repo)

git init

# Create and switch to main branch

git checkout -b main

# Set remote (remove existing origin first if necessary)

git remote remove origin 2>$null

git remote add origin https://Liberty666-666:ghp\_0Hi6EhZAhX301v4Cg26ebiRLBkgqAU1MbOv1@github.com/Liberty666-666/CufflessBPNet-Development-of-Neural-Network-Model-for-Cuffless-Blood-Pressure-Estimation.git

# Stage and commit all files

git add .

git commit -m "Initial commit after cleaning files"

# Push to GitHub

git push -u origin main

the red note is classic token, which is obtained in developer mode