**Report**

**112/09/13 進度報告 張家菖**

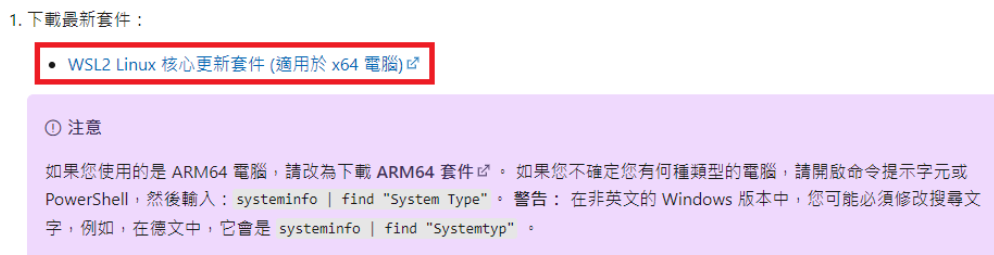
WSL2:  
到 控制台>程式集>程式與功能，點選左邊的「開啟或關閉Windows功能」，然後將紫色部分前面的框框打勾，接著重新啟動電腦

一張含有 文字, 電子產品, 螢幕擷取畫面, 陳列 的圖片

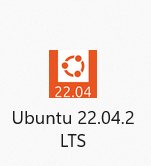
自動產生的描述  
終端機(管理員):

dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart

下載 Linux 核心更新套件



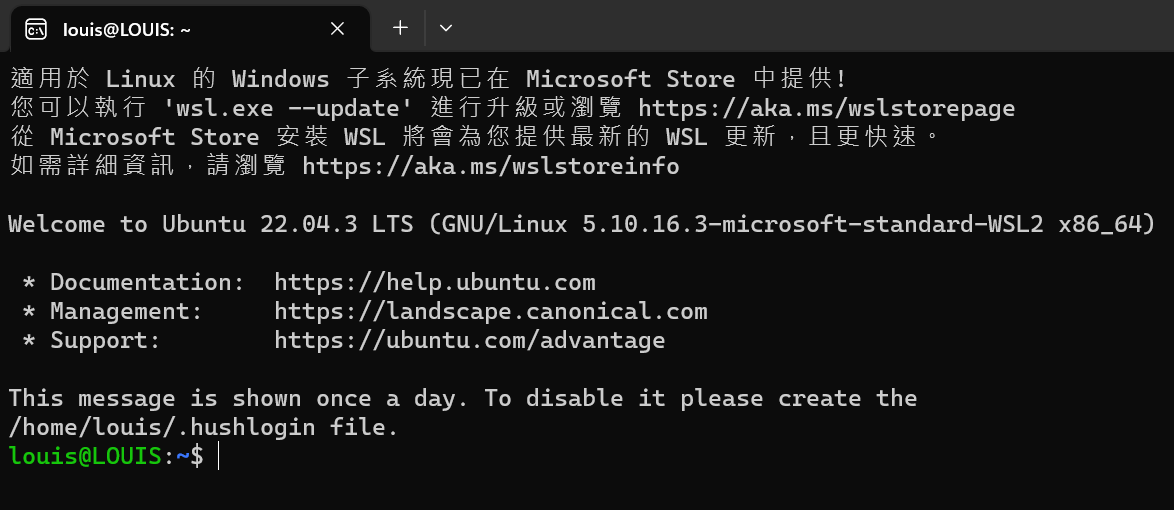
Microsoft store安裝



將 WSL2 設定為預設版本

wsl --set-default-version 2

啟動WSL2



sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libsqlite3-dev libreadline-dev libffi-dev wget libbz2-dev

## (python3.7相依庫)

sudo add-apt-repository ppa:deadsnakes/ppa

## (Press [ENTER] to continue or Ctrl-c to cancel adding it.)

sudo apt install python3.7

sudo apt-get update -y

sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 1

sudo update-alternatives --config python3

## (替換成python3.7)

sudo apt install python3-pip

sudo apt install python3.7-distutils

python3 -m pip install --upgrade pip

pip install tensorflow==1.14.0

pip install ray==0.7.6

pip install gym==0.17.3

pip install opencv-python

pip install setproctitle

pip install tabulate

pip install requests

pip install psutil

pip install lz4

pip uninstall protobuf

pip install protobuf==3.19.0

pip install opencv-python-headless

## (libGL.so.1報錯解決方法)

git clone https://github.com/JiaChangGit/NeuroCuts\_robin.git

cd NeuroCuts\_robin/

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

(ipc2\_100k訓練結果)

## (後續指令全部變成python3 且加 . )

EX:

python run\_neurocuts.py --rules=ipc2\_100k --dump-dir=/content/drive/MyDrive/CIAL/neurocut\_update\_efficut/w=1,u=0/ipc2\_100k --depth-weight=1 --partition-mode=efficuts

變成

python3 run\_neurocuts.py --rules=ipc2\_100k --dump-dir=./content/drive/MyDrive/CIAL/neurocut\_update\_efficut/w=1,u=0/ipc2\_100k --depth-weight=1 --partition-mode=efficuts

EX:

pkl檔轉換成txt檔

python pickle-to-json.py [.pkl路徑]

改成:

python3 pickle-to-json.py [~/… .pkl路徑]



轉換pkl成test(txt)，test(txt)內容

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

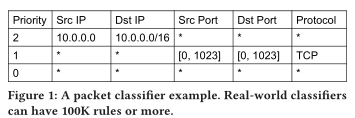
自動產生的描述

**Report**

**112/09/06 進度報告 張家菖**

1. **交接-概述**
2. **Packet Classification**

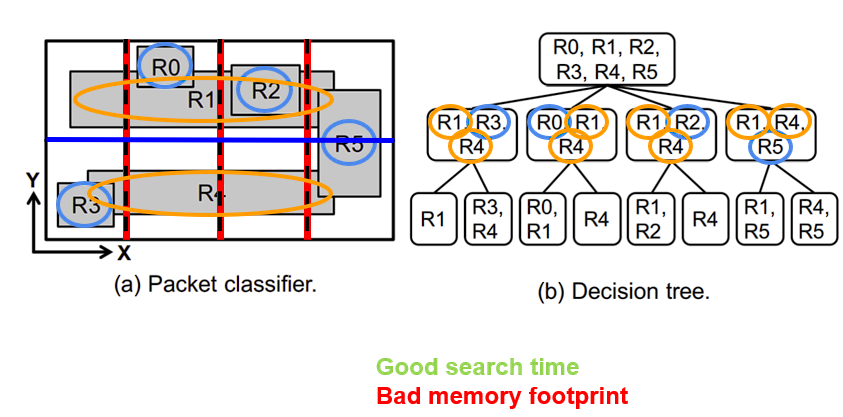
Packet classification類似於多維空間中的點位置問題。一個數據包被表示為這個空間中的一個點。我們要找到所有（包含數據包對應的）點的位置。如下圖(Fig1)，有3個rules和5個維度。



兩種常見的方法來構建決策樹進行Packet Classification:

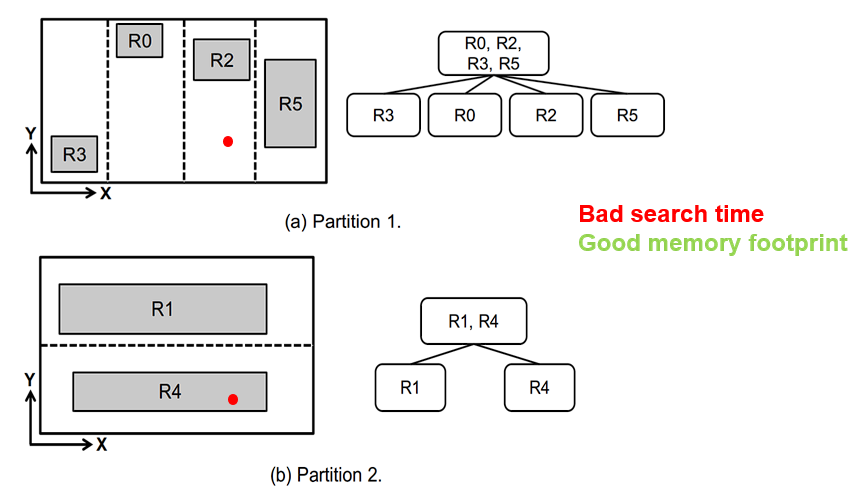
* 1. Node cutting (Cutting):

The main idea is to split nodes in the decision tree by cutting them along dimensions.



* 1. Rule partition (Grouping):

In particular, if a rule has a large size along one dimension, cutting along that dimension will result in that rule being added to many nodes. Rule replication can lead to decision trees with larger depths and sizes, which translate to higher classification time and lower memory footprint.



1. **Neural Packet Classification**
   1. 為什麼選擇RL?

構建決策樹是從一個節點開始切割它，這使得我們在完成構建之前，不知道這個切分的決定是否是一個好的決定，而這符合RL的特性，如同alpha Go下圍棋一樣，在遊戲結束前，無法得知這一步棋的好壞，在決策樹建立完前，無法得知樹的好壞。

Building a decision tree can be easily cast as an RL problem: the environment’s state is the current decision tree, an action is either cutting a node or partitioning a set of rules, and the reward is either the classification time, memory footprint, or a combination.

* 1. Reward Function:

一張含有 文字, 字型, 螢幕擷取畫面, 收據 的圖片

自動產生的描述

Cutting: 因為最差會是在children nodes最多的地方，所以算children(MAX)。Memory size則是所有children nodes。

Grouping: 因為rule只會在一個tree中，所以要找所有樹的children node(SUM)。Memory size則是所有children nodes。

一張含有 文字, 螢幕擷取畫面, 圖表, 字型 的圖片

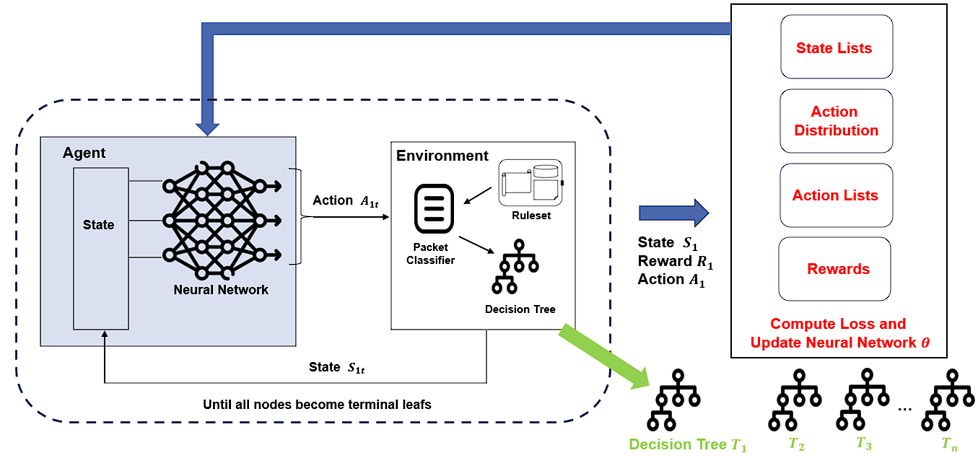
自動產生的描述一張含有 文字, 螢幕擷取畫面, 圖表, 字型 的圖片

自動產生的描述

Tn: search time; Sn: memory size

* 1. 計算Loss:

State Lists 、Action Distribution、Action Lists、Rewards，透過這四個資訊來計算Loss。

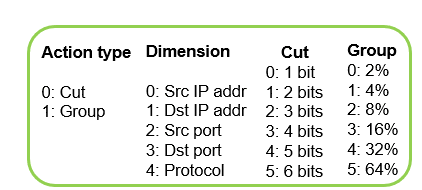
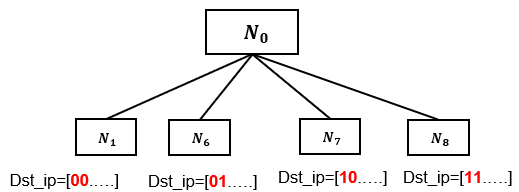


State Lists: 如下圖，分別是source ip (min MAX); destination ip (min MAX); source port (min MAX); destination port (min MAX); protocol (min MAX)。



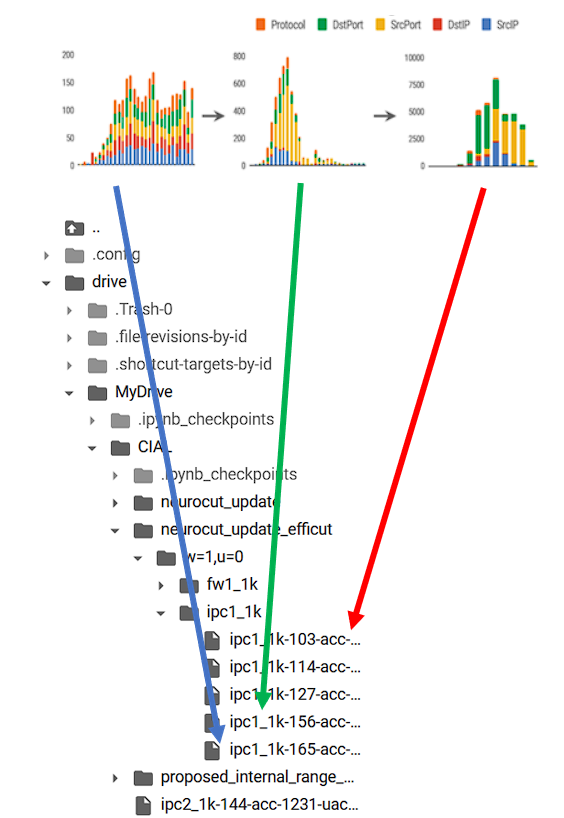
Action Distribution: Neural network給的action 是一個機率分布(RL具有隨機性)。

Action Lists: 如下圖，舉例: 𝑨(𝟎 )= [0,1,1]代表，依據destination ip做2-bits cutting。

訓練過程:

如下圖，決策樹會愈來愈好，可藉由result.json來判斷收斂。



1. **程式**

環境 :

1. Colab，掛接雲端硬碟，目的是為了讓訓練的結果，能夠存放在自己的雲端。

# 此內容會顯示為程式碼

from google.colab import drive

drive.mount('/content/drive')

1. 將python降版至3.7，安裝tool，並且clone github

# 此內容會顯示為程式碼

#-----To install specific version 3.7 of python in Google Colab-----

!sudo apt-get install python3.7

!sudo apt-get update -y

!sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.7 1

!sudo update-alternatives --config python3 #按有注音的數字鍵(選3.7)

!python --version

!sudo apt install python3-pip

!sudo apt install python3.7-distutils

!python -m pip install --upgrade pip

!pip install tensorflow==1.14.0

!pip install ray==0.7.6

!pip install gym==0.17.3

!pip install opencv-python

!pip install setproctitle

!pip install tabulate

!pip install requests

!pip install psutil

!pip install lz4

!pip uninstall protobuf

!pip install protobuf==3.19.0

# 歐學長的github，亦可改成學生的: https://github.com/JiaChangGit/NeuroCuts\_robin

!git clone https://github.com/ourobin2/NeuroCuts\_robin.git

%cd NeuroCuts\_robin/

%ls

1. 指令說明 --rules=[要訓練的ruleset] --dump-dir=[訓練好的tree要存的路徑] --depth-weight=[w的值，預設為1] --partition-mode=efficuts[root node模式，預設為None]

**(若欲運行版本3、版本4，需更改程式碼)**

**版本(1) proposed:**

# 此內容會顯示為程式碼

#example

!python run\_neurocuts.py --rules=ipc2\_100k --dump-dir=/content/drive/MyDrive/CIAL/proposed\_internal\_range\_plus1\_noefficut/w=0.99999,u=0/ipc2\_100k --depth-weight=0.99999

版本(2) proposed+efficut:

# 此內容會顯示為程式碼

#example

!python run\_neurocuts.py --rules=ipc2\_100k --dump-dir=/content/drive/MyDrive/CIAL/neurocut\_update\_efficut/w=1,u=0/ipc2\_100k --depth-weight=1 --partition-mode=efficuts

版本(3) efficut:

# 此內容會顯示為程式碼

#example

!python run\_neurocuts.py --rules=ipc2\_100k --dump-dir=/content/drive/MyDrive/CIAL/neurocut\_update\_efficut/w=1,u=0/ipc2\_100k --depth-weight=1 --partition-mode=efficuts

版本(4) Neurocuts:

# 此內容會顯示為程式碼

#example

!python run\_neurocuts.py --rules=ipc2\_100k --dump-dir=/content/drive/MyDrive/CIAL/neurocut\_update/w=1,u=0/ipc2\_100k --depth-weight=1

1. 查看Data:

用ipc2\_1k當ruleset，memory access: 144(Search time), update memory access: 1231(Update time), bytes\_per\_rule: 121.79(memorySize / rule)。

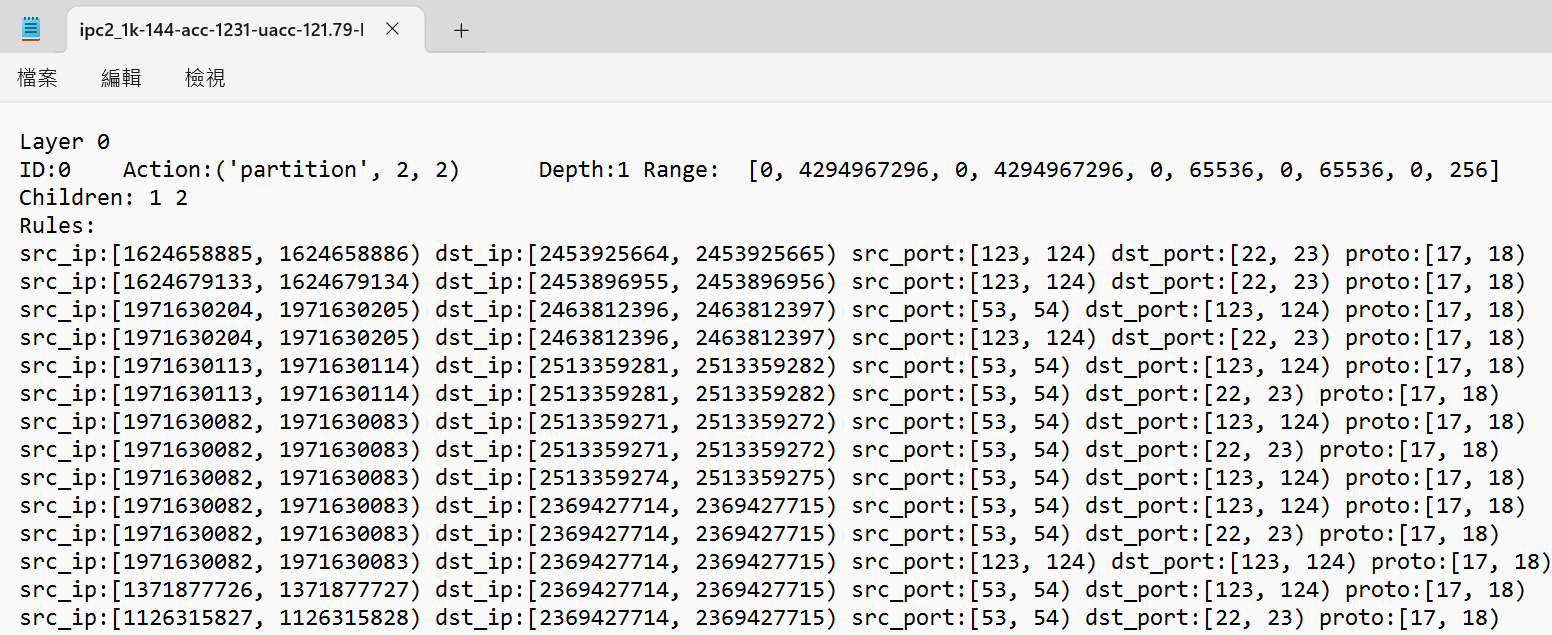


pkl檔轉換成txt檔:

# 此內容會顯示為程式碼

!python pickle-to-json.py /content/drive/MyDrive/CIAL/proposed\_internal\_range\_plus1\_noefficut/w=0.99999,u=0/acl5\_100k/acl5\_100k-6-acc-15594-uacc-17.08-bytes-1688327649.3373015.pkl





想要查看tree的架構可以使用 inspect\_tree.py 指令:

# 此內容會顯示為程式碼

!python inspect\_tree.py /content/drive/MyDrive/CIAL/proposed\_internal\_range\_plus1/w=0.99999,u=0/acl2\_100k/acl2\_100k-55-acc-17590-uacc-27.37-bytes-1688528736.69055.pkl

查看訓練狀況是否收斂:

可以至vm查看訓練詳細資料。

舉例路徑: /root/ray\_results/neurocuts\_efficuts/PPO\_tree\_env\_0\_rules=\_content\_NeuroCuts\_robin\_ruleset\_fw5\_100k,tree\_gae\_lambda=0.95\_2023-07-12\_05-57-30ab7gapzh/result.json



可判斷收斂狀況，如下圖，未收斂，因為node\_remaining\_min != 0 ，用來判斷是本次訓練較慢，還是colab途中提前斷掉。

(註node\_remaining\_min = rules\_remaining\_min + 1)



Cycle time測試:

1. pkl檔轉換成txt檔:

# 此內容會顯示為程式碼

!python pickle-to-json.py /content/drive/MyDrive/CIAL/proposed\_internal\_range\_plus1\_noefficut/w=0.99999,u=0/acl5\_100k/acl5\_100k-6-acc-15594-uacc-17.08-bytes-1688327649.3373015.pkl

1. 會在/content/NeuroCuts\_robin 路徑下產生test.txt，移至run cycle time的環境(178)
2. 編譯+執行 ./partition [ruleset] [.txt] [trace]:

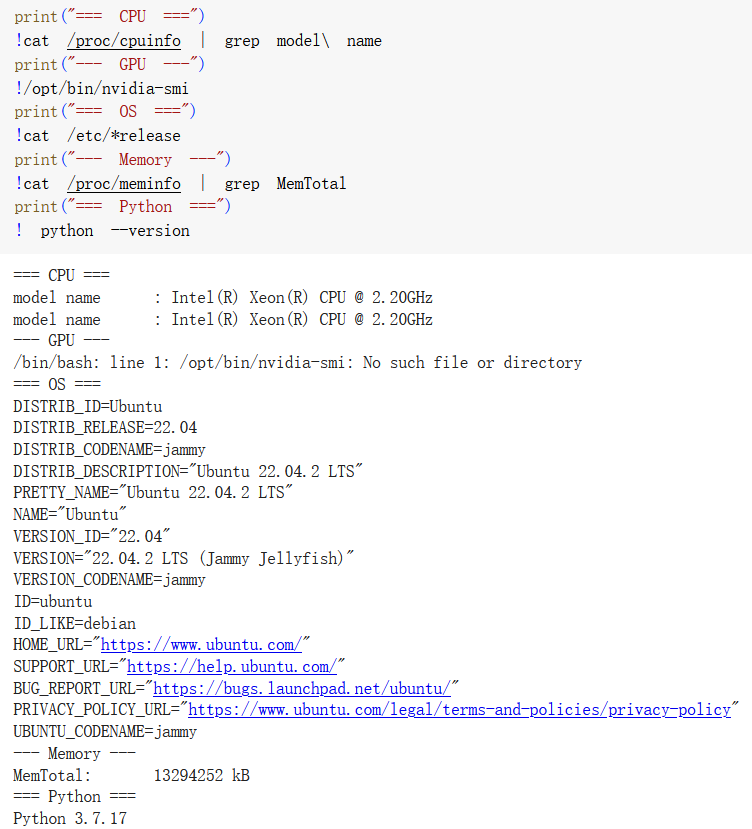
# 此內容會顯示為程式碼

//example

gcc partition\_time.c -lm -o partition

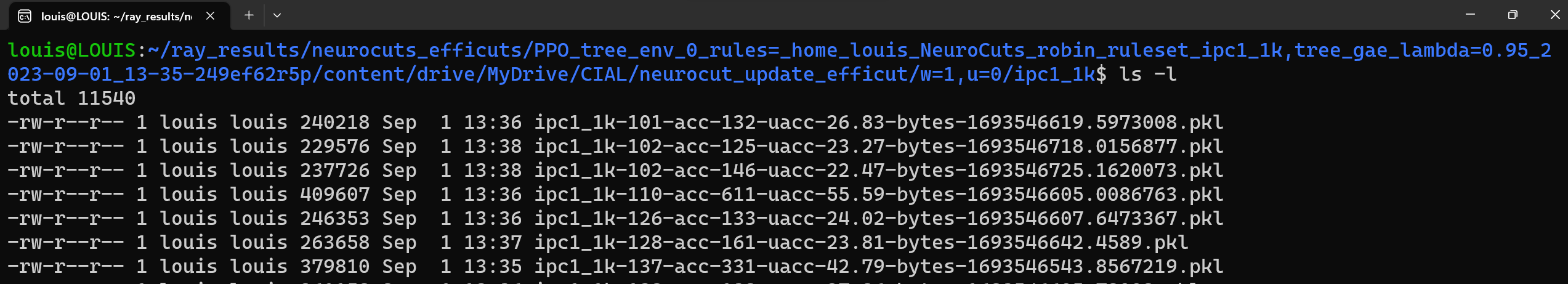
./partition ruleset/acl1\_100k tree/largeness\_acl1\_100k\_41\_21.99.txt trace/acl1\_100k\_trace

學生Colab配置:



**五、WSL2-Ubuntu22.04.2架設**

以實現，如下圖:

****

問題: 無法更改pkl結果存處根路徑。 --dump-dir=[訓練好的tree要存的路徑]

更改路徑，還是會存在同一個根位置

(~/ray\_results/neurocuts\_efficuts/PPO\_tree\_env\_0\_rules=\_home\_louis\_NeuroCuts\_robin\_ruleset\_ipc1\_1k,tree\_gae\_lambda=0.95\_2023-09-01\_13-35-249ef62r5p/content/drive/MyDrive/CIAL/neurocut\_update\_efficut/w=1,u=0/ipc1\_1k)