

# AI如何學習-硬體流程

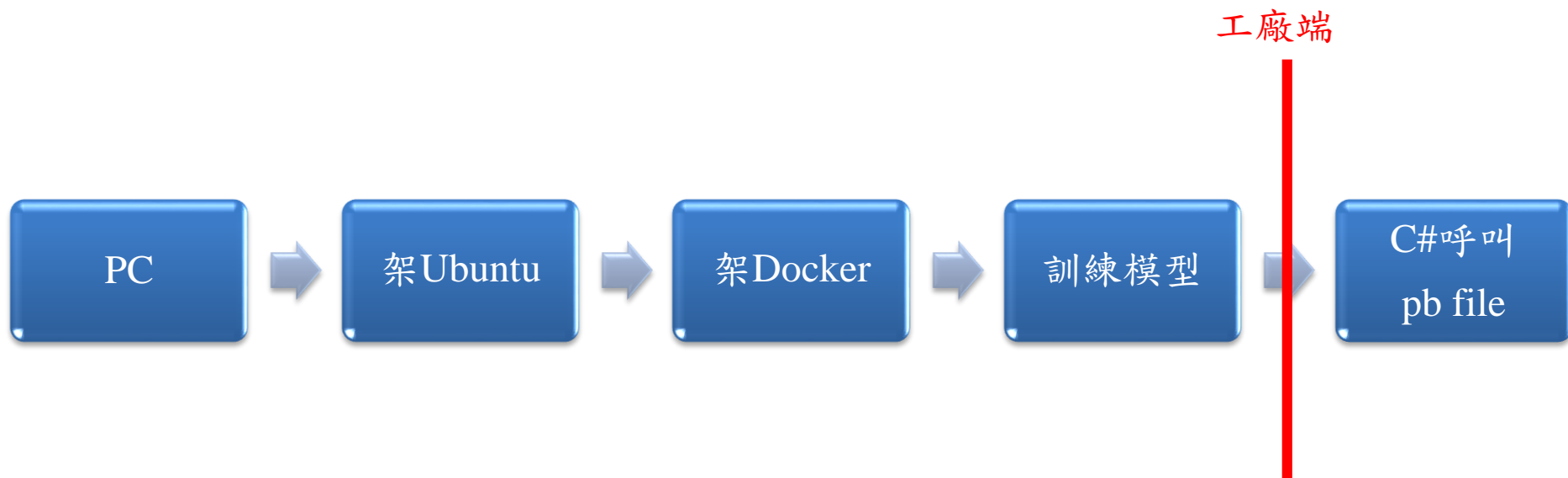
Allen

2019/11/08

# 報告內容

- 前置作業
- 程式執行
- 工廠運作

# 總體流程



# 前置作業

1. 準備 USB



2. 下載Ubuntu 18.04的iso檔

3. 使用isotousb的軟體燒錄

([http://201708.mediafire.com/file/6r4syo2c4u8dhsb/ISOtoUSBPortable\\_1.5\\_azo.exe](http://201708.mediafire.com/file/6r4syo2c4u8dhsb/ISOtoUSBPortable_1.5_azo.exe))

注意：燒錄時會整個格式化，請先備份重要檔案

4. 重開機進入bios，改boot option優先啟用USB即可重開

5. 進入ubuntu安裝環節，建議將系統與/home分割出來，再來輸入帳號密碼即可

# 前置作業

1. 從<https://www.nvidia.co.jp/Download/index.aspx>找到”符合該硬體”的驅動下載
2. 打開terminal (Ctrl + Alt + T)，找到剛才載驅動的位置

```
sudo bash NVIDIA-Linux-x86_64-430.50.run
```

(中間如果有缺gcc套件等等，請再用sudo apt-get install指令來下載)

(注意，如果未來驅動不見了，在選擇kernel是否要跟換，要選”否”)

3. 輸入nvidia-smi，會出現下列圖示的話，代表驅動有灌成功

NVIDIA-SMI 430.50				Driver Version: 430.50			CUDA Version: 10.1		
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC			
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.			
0	GeForce GTX 108...	On	00000000:17:00.0	Off		N/A			
41%	71C	P2	92W / 300W	4647MiB / 11178MiB	40%	Default			
1	GeForce GTX 108...	On	00000000:65:00.0	Off		N/A			
33%	28C	P8	22W / 320W	0MiB / 11176MiB	0%	Default			

# 前置作業

注意：由於會使用docker，因此不介紹如何灌cuda

1. 進入docker網址的教學<https://docs.docker.com/install/linux/docker-ce/ubuntu/>
2. 打開terminal

---

```
sudo apt-get update -y
```

```
sudo apt-get install \  
  apt-transport-https \  
  ca-certificates \  
  curl \  
  gnupg-agent \  
  software-properties-common
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

# 前置作業

```
sudo apt-key fingerprint 0EBFCD88
```

```
sudo add-apt-repository \  
"deb [arch=amd64] https://download.docker.com/linux/ubuntu \  
$(lsb_release -cs) \  
stable"
```

```
sudo apt-get update  
sudo apt-get install docker-ce docker-ce-cli containerd.io  
apt-cache madison docker-ce
```

以上，結束docker安裝

下一頁將介紹nvidia-docker安裝

# 前置作業

(nvidia docker安裝指令網址：<https://github.com/NVIDIA/nvidia-docker>)

---

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
```

```
curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key add -
```

```
curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sudo tee  
/etc/apt/sources.list.d/nvidia-docker.list
```

(此為一行指令，勿分開輸入)

```
sudo apt-get update && sudo apt-get install -y nvidia-container-toolkit
```

```
sudo systemctl restart docker
```



# 前置作業

1. 打開terminal
2. 做好的docker下來跑，指令如下：

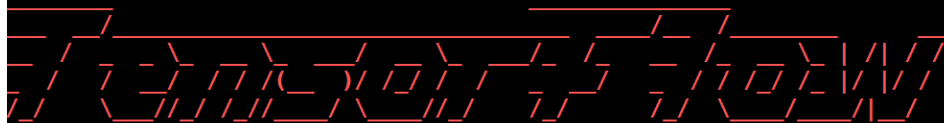
```
sudo docker run -it --gpus all --rm -e DISPLAY=$DISPLAY -v  
$HOME/.Xauthority:/root/.Xauthority -v /home/user:/workspace --network=host --  
name=VAETEST markliou/python3-tensorflow-gpu bash  
(此為一行指令，勿分開輸入)
```

/home/user：最好是設在當前資料夾，裡面應該要包含py檔以及dataset

VAETEST：此為指定container的名字，如有再開一個container會需要修改名稱

成功畫面如下：

```
(base) allen@server-X299-UD4:~$ sudo docker run -it --gpus all --rm -e DISPLAY=$DISPLAY -v $HOME/.Xauthority:/root/.Xauthority -v /home/allen:/workspace --network=host --name=TEST markliou/python3-tensorflow-gpu bash
```



```
WARNING: You are running this container as root, which can cause new files in mounted volumes to be created as the root user on your host machine.
```

```
To avoid this, run the container by specifying your user's userid:
```

```
$ docker run -u $(id -u):$(id -g) args...
```

```
root@server-X299-UD4:/workspace#
```

# 執行作業

1. 將寫好的程式檔(.py)，放到剛剛container設定好的路徑底下
2. 一般根據我的習慣，會將main檔和model檔分開寫，並且在訓練的過程中，順便做inference (大概每5個epoch就會做一次)
3. epoch：全部資料要餵到模型的次數
4. batch size：一次餵到模型資料數量
5. step (又稱iteration)：餵一個batch size的資料即為一次

例：

全部資料筆數=1000

batch size = 40

因此，需要跑step=25

而當這1000筆資料，模型都看過的時候稱為1個epoch

1. 在上述訓練完之後，會將pb檔留下來，其中pb檔會包含計算圖和全重
2. 利用C#去call pb檔，其中也要包含整個tensorflow的 package
3. 並且還需要根據後面有python opencv的程序，再去做撰寫



# Thank You !



責任 **R**esponsibility

創新 **I**nnovation

卓越 **S**uperiority

開創 **E**ntrepreneurship

[www.coretronic.com](http://www.coretronic.com)