MNIST TUTORIAL

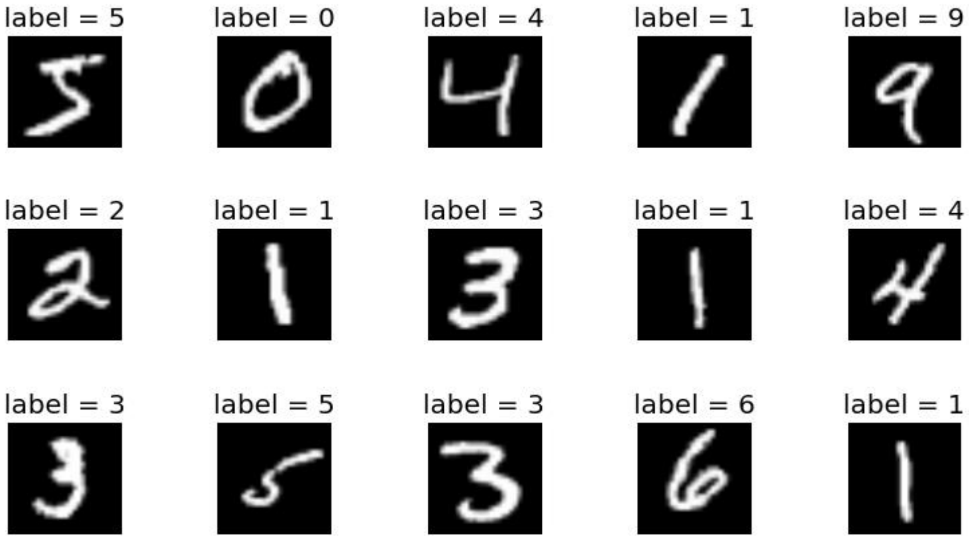
N26091194 鄧立昌

OUTLINE

- MNIST dataset
- Convolution
- Pooling
- Flatten
- Full connected layer
- Activation function
- Loss function
- Optimizer
- Install pytorch(GPU)
- DL PROJECT STRCTURE

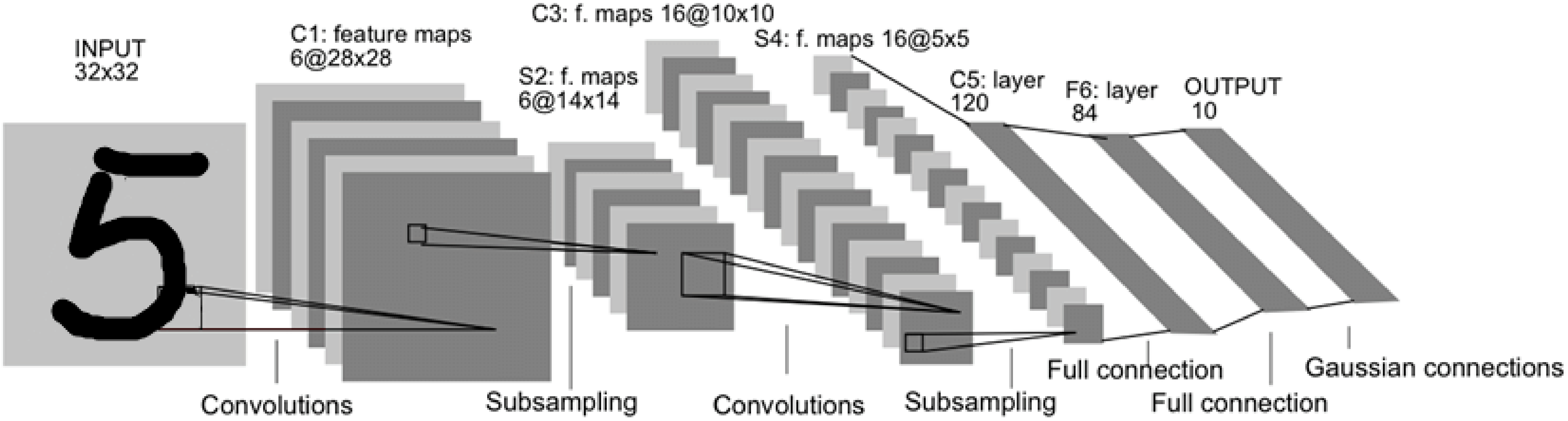
MNIST DATASET

- 手寫數字辨識資料 http://yann.lecun.com/exdb/mnist/
- Deep learning 的 hello world
- 由卷積網絡之父 Yann LeCun 蒐集建立
  - 與 Geoffrey Everest Hinton / Yoshua Bengioej 共同獲得 2018年圖靈獎 貢獻: 深度學習



LENET-5

- GradientBased Learning Applied to Document Recognition - Yann LeCun et al.



An early (Le-Net5) Convolutional Neural Network design, LeNet-5, used for recognition of digits

Source

FREE GPU PLATFORM

Colab

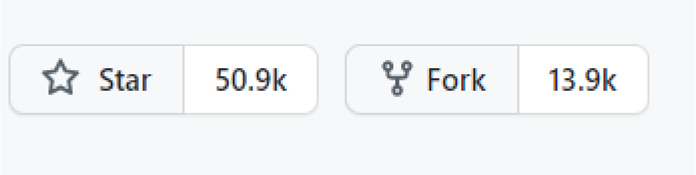
- Pros
  - 可跟雲端硬碟同步
- Cons
  - GPU 是隨機分配的
  - 使用量限制是動態的 (沒有明確定義)
  - 容易斷線(使用量超過限制)

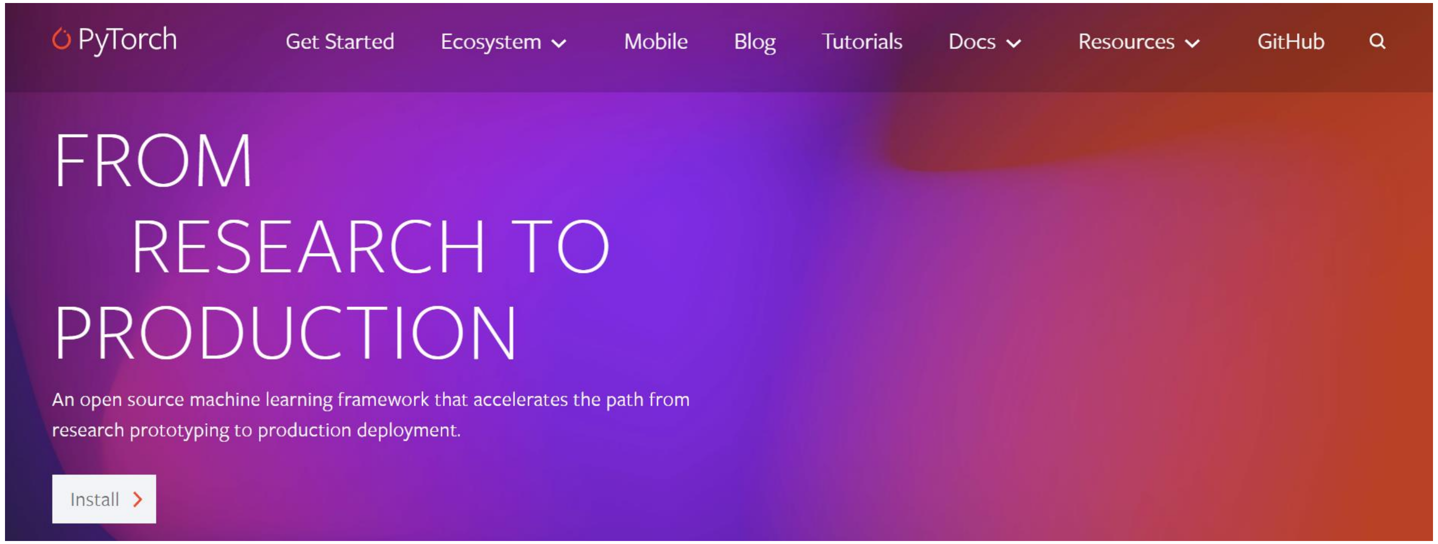
Kaggle

- Pros
  - 較穩定，不易斷線
  - GPU: Tesla P100 (30hrs/week)
- Cons
  - 上傳資料較麻煩

PYTORCH

- Pros
  - 高度自訂性
  - 龐大的社群 / 討論區
  - 好上手 (相較於 Tensorflow)
- Cons
  - 難上手 (相較於 keras)





FROM RESEARCH TO PRODUCTION

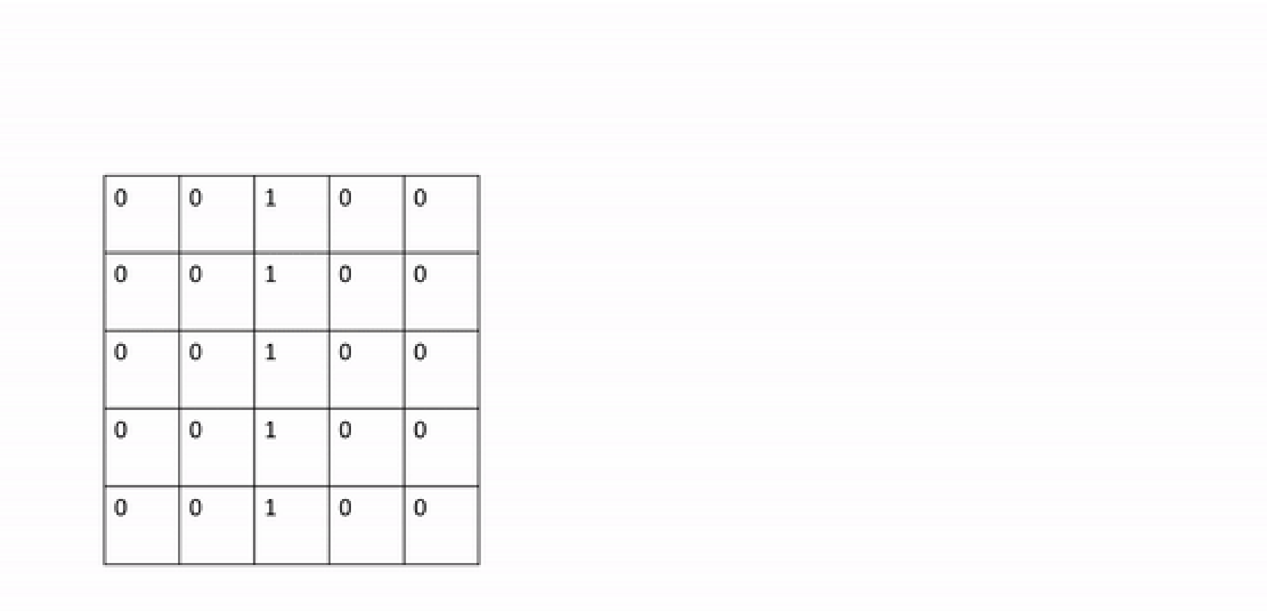
An open source machine learning framework that accelerates the path from research prototyping to production deployment.

CONVOLUTION

- Kernel: 用來擷取圖片區域特徵 e.g. 邊緣、水平線、垂直線、...
  - Stride: 決定步伐大小
  - Zero padding: 是否使用0來對周圍進行填充

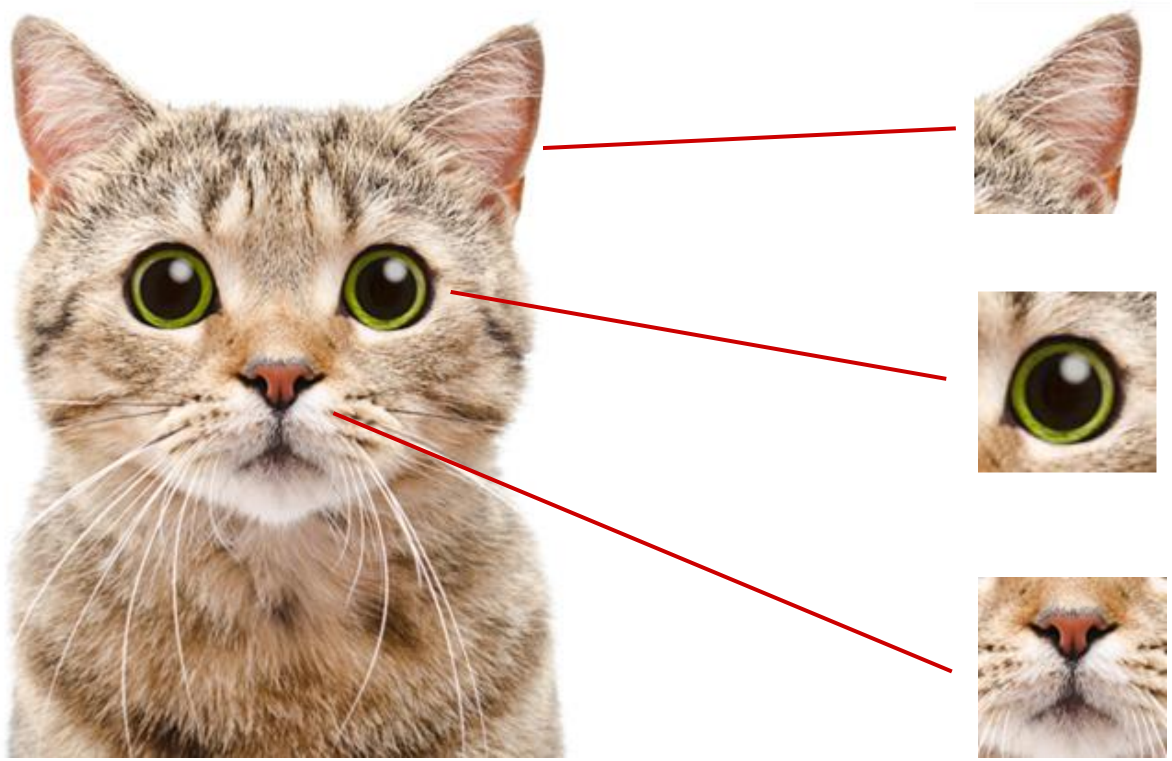
1	0	-1
2	0	-2
1	0	-1

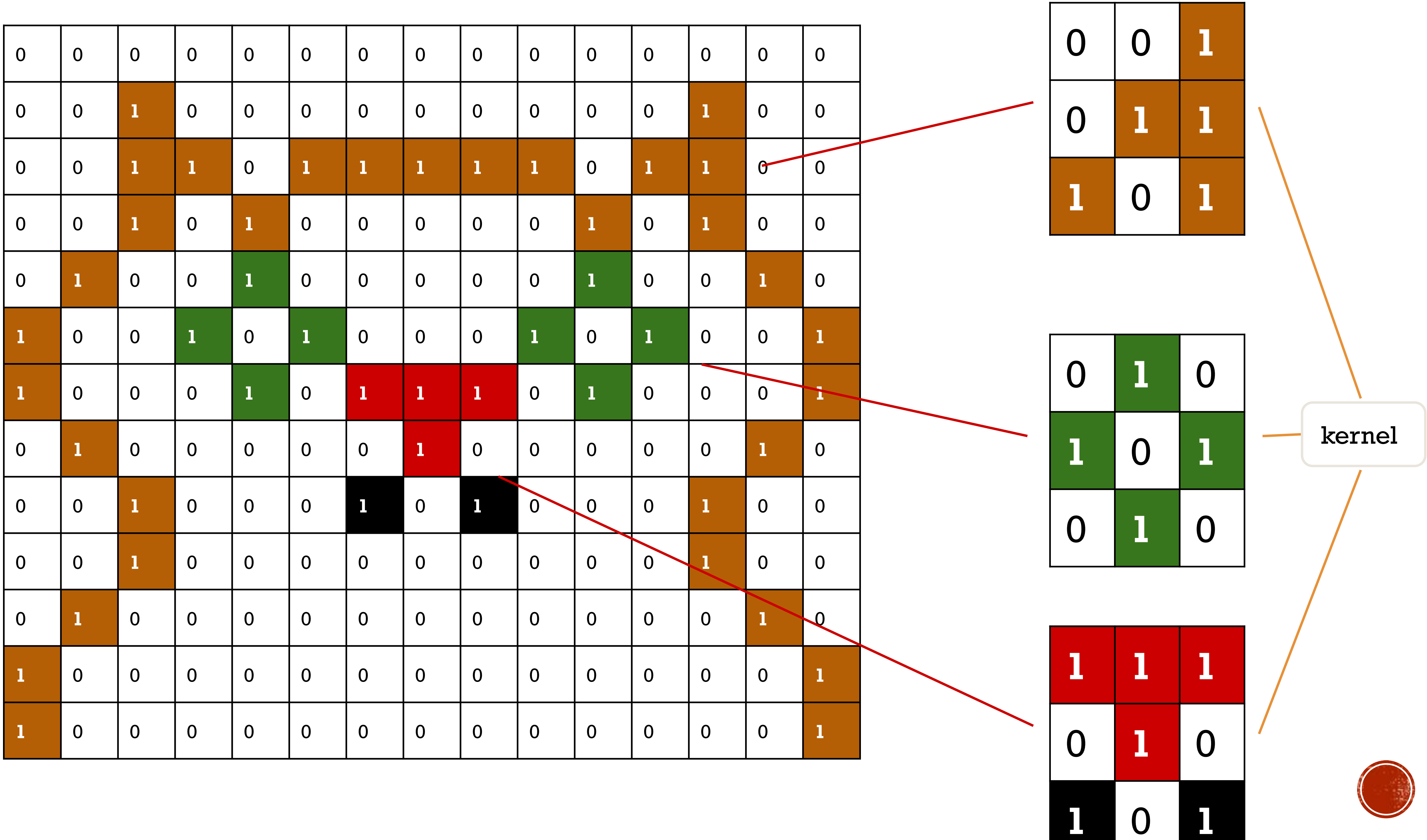
kernel



0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0

CONVOLUTION(CONT.)





0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	1	1	1	1	1	0	1	1	0	0
0	0	1	0	1	0	0	0	0	0	1	0	1	0	0
0	1	0	0	1	0	0	0	0	0	1	0	0	1	0
1	0	0	1	0	1	0	0	0	1	0	1	0	0	1
1	0	0	0	1	0	1	1	1	0	1	0	0	0	1
0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	1	0	0	0	1	0	1	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	1

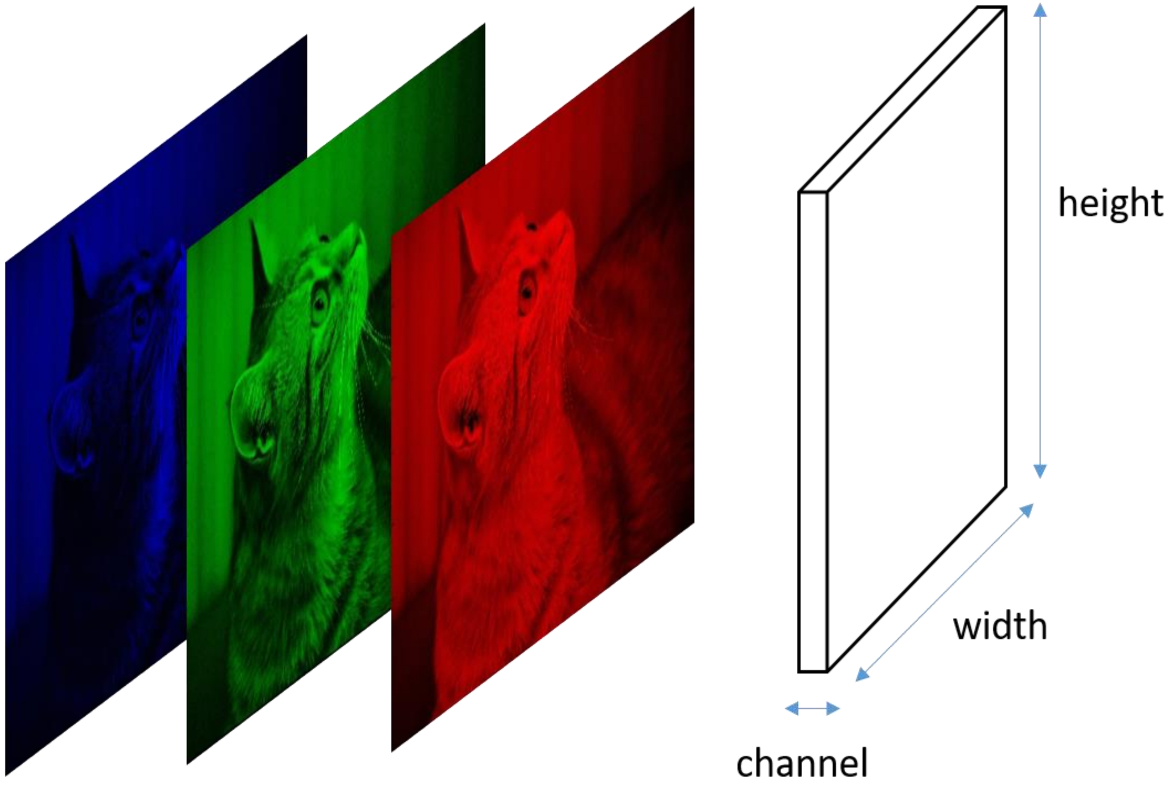
0	0	1
0	1	1
1	0	1

0	1	0
1	0	1
0	1	0

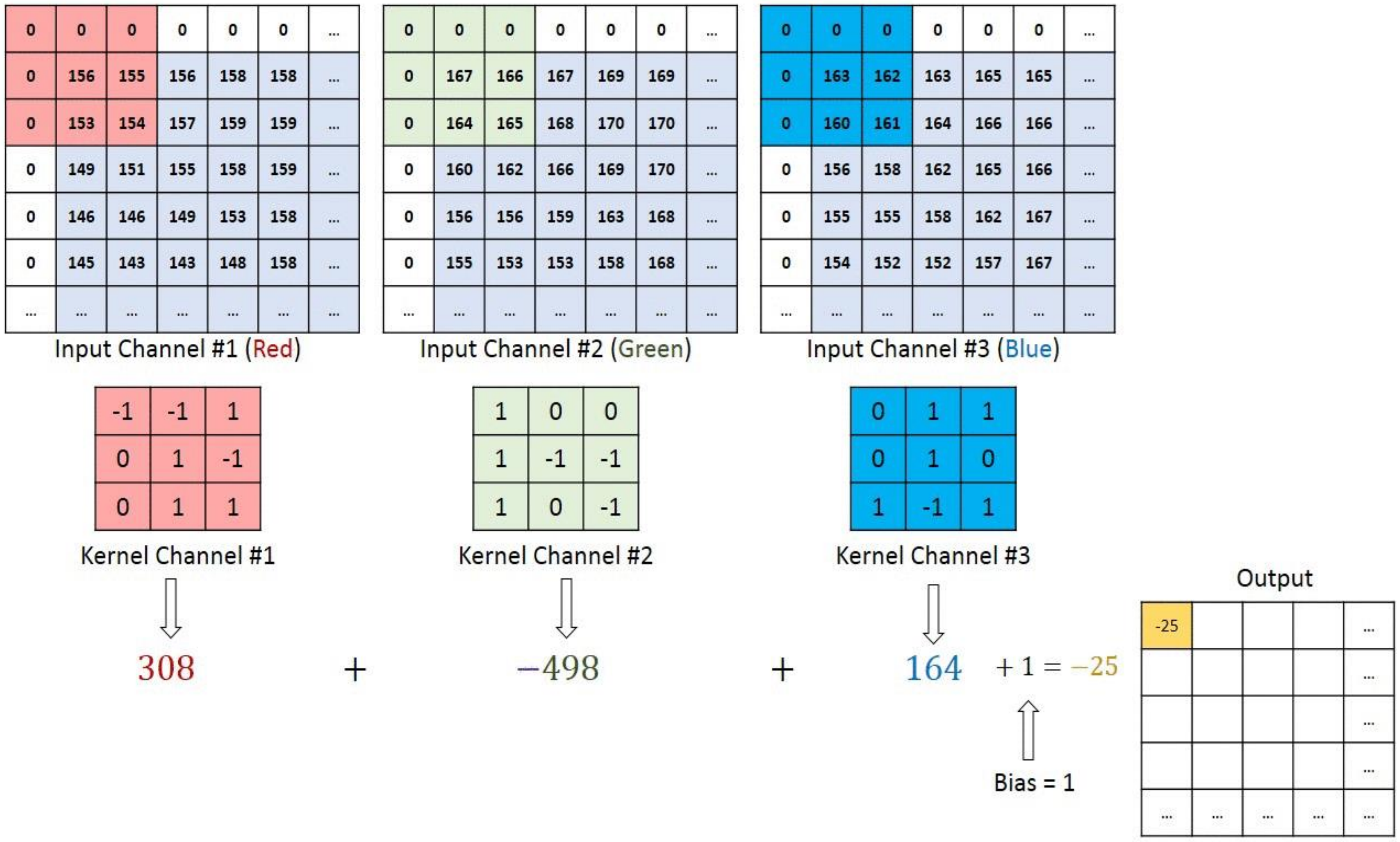
1	1	1
0	1	0
1	0	1

kernel

RGB CHANNEL



RGB CHANNEL(CONT.)



0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #2

0	1	1
0	1	0
1	-1	1

Kernel Channel #3

$$308 \quad + \quad -498 \quad + \quad 164 \quad +1 = -25$$

Bias = 1

Output

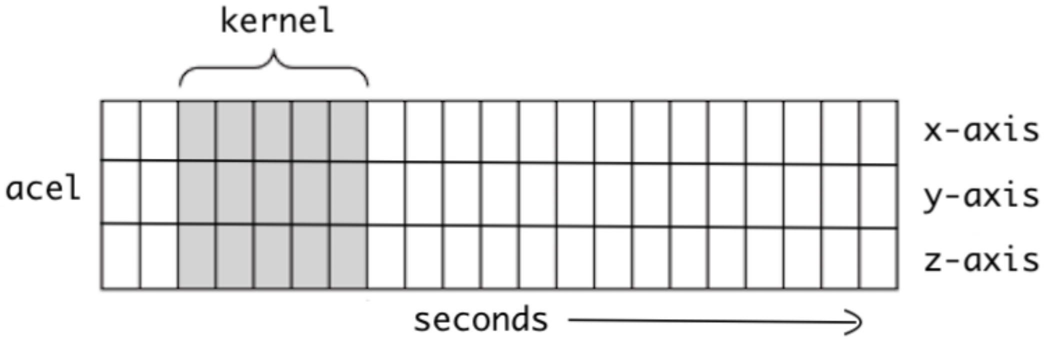
-25				...
				...
				...
				...
...	...	...	...	...

CONVOLUTION LAYER 程式碼

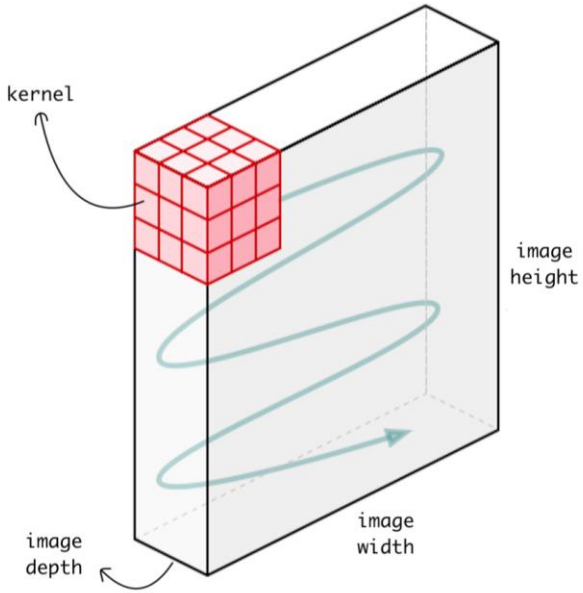
- torch.nn.Conv2d(in_channels, out_channels, kernel_size, padding)
  - in_channels: 輸入幾個 channel ex. RGB就是三個channel
  - out_channels: 要輸出幾個 channel
  - kernel_size: filter (feature detection) 要 n*n
  - padding (補 0): 控制 zero-padding 的數目

補充

- https://stackoverflow.com/questions/42883547/intuitive-understanding-of-1d-2d-and-3d-convolutions-in-convolutional-neural-n



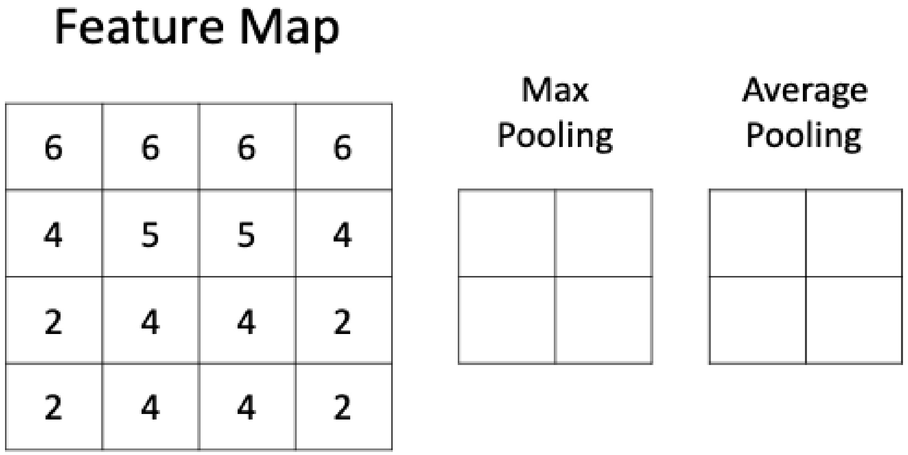
1D convolution



2D convolution

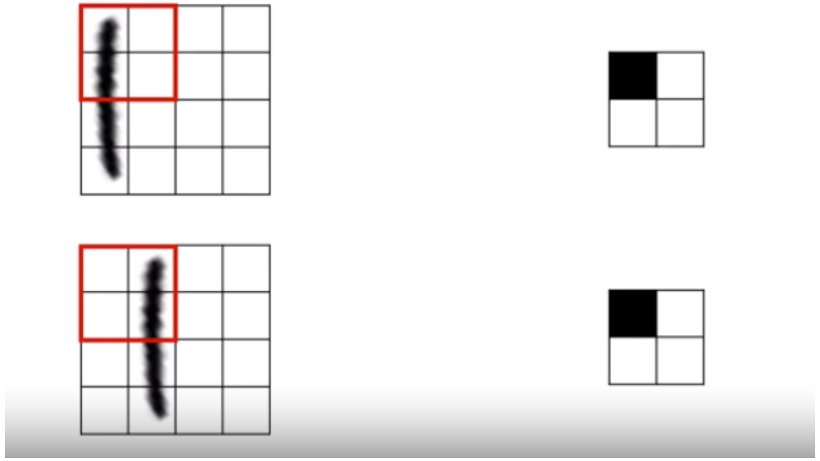
POOLING

- Pooling:
  - 1. 減少參數量跟計算
  - 2. pooling 把 feature map 的區域資訊做整合，讓模型更 robust



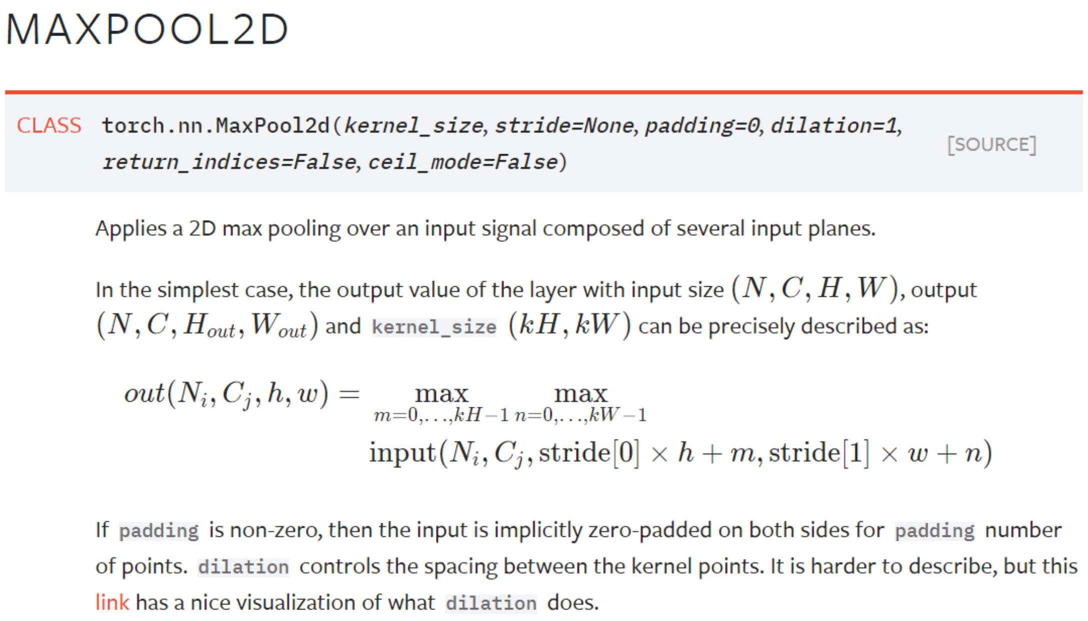
Feature Map

6	6	6	6
4	5	5	4
2	4	4	2
2	4	4	2



POOLING LAYER 程式碼

- torch.nn.MaxPool2d(kernel_size=2)
- torch.nn.AvgPool2d(kernel_size=2)



MAXPOOL2D

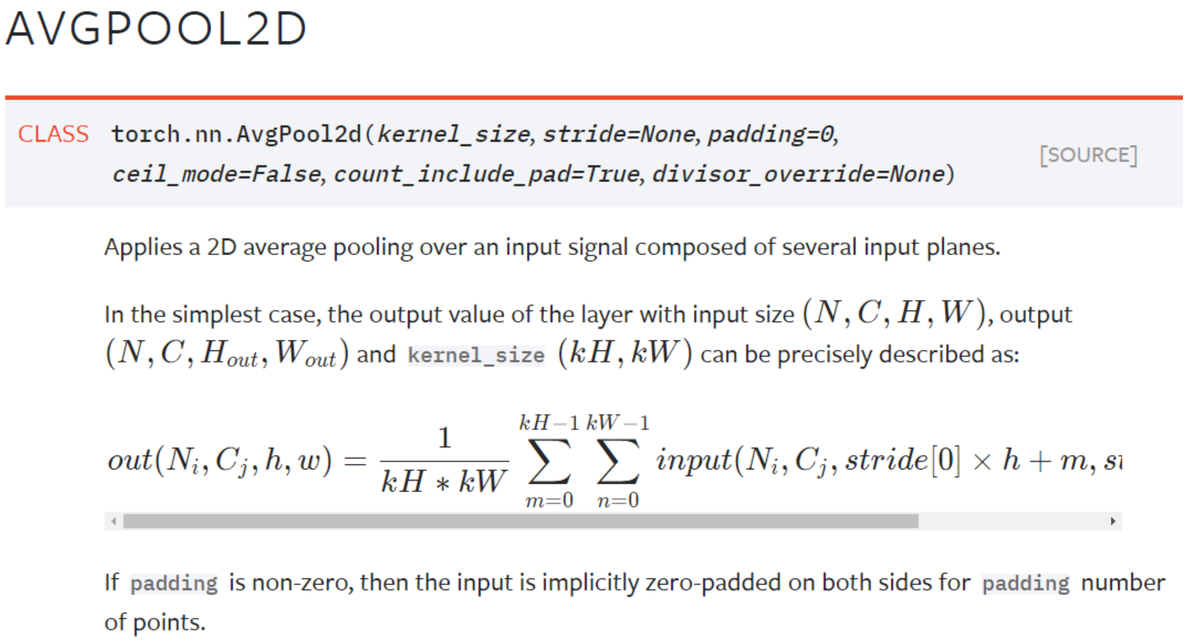
CLASS `torch.nn.MaxPool2d(kernel_size, stride=None, padding=0, dilation=1, return_indices=False, ceil_mode=False)` [SOURCE]

Applies a 2D max pooling over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C, H, W), output (N, C, H_{out}, W_{out}) and `kernel_size` (kH, kW) can be precisely described as:

$$out(N_i, C_j, h, w) = \max_{m=0,\ldots,kH-1} \max_{n=0,\ldots,kW-1} \text{input}(N_i, C_j, \text{stride}[0] \times h + m, \text{stride}[1] \times w + n)$$

If `padding` is non-zero, then the input is implicitly zero-padded on both sides for `padding` number of points. `dilation` controls the spacing between the kernel points. It is harder to describe, but this link has a nice visualization of what `dilation` does.



AVGPOOL2D

CLASS `torch.nn.AvgPool2d(kernel_size, stride=None, padding=0, ceil_mode=False, count_include_pad=True, divisor_override=None)` [SOURCE]

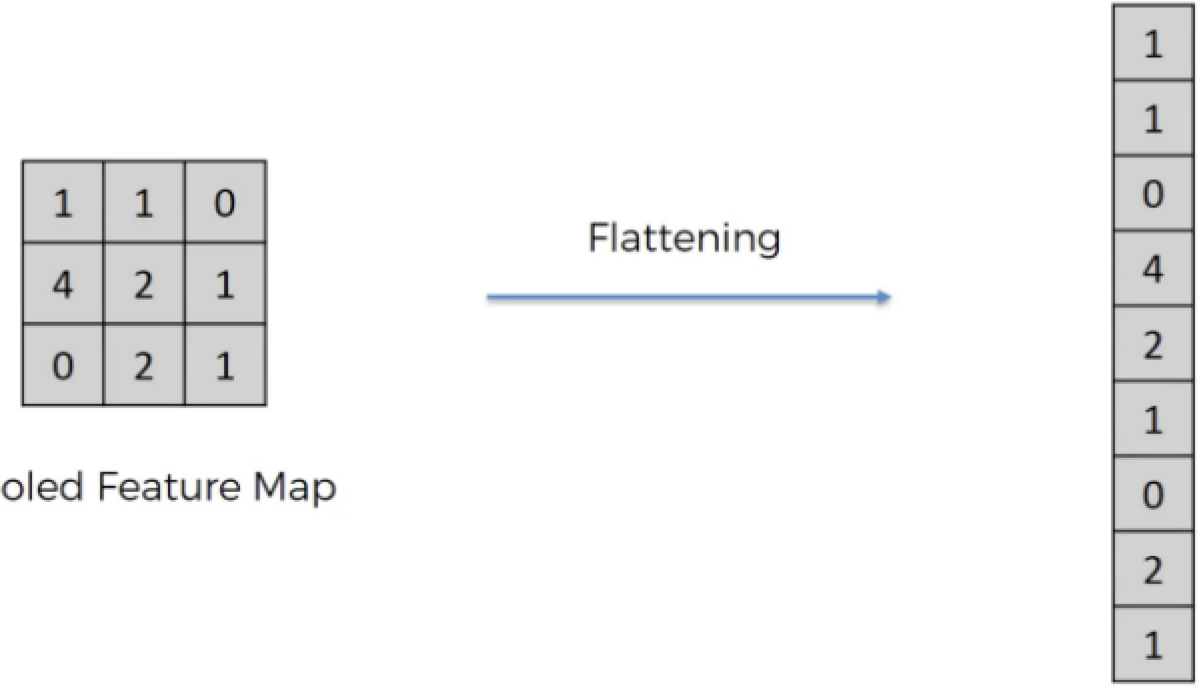
Applies a 2D average pooling over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C, H, W), output (N, C, H_{out}, W_{out}) and `kernel_size` (kH, kW) can be precisely described as:

$$out(N_i, C_j, h, w) = \frac{1}{kH * kW} \sum_{m=0}^{kH-1} \sum_{n=0}^{kW-1} input(N_i, C_j, stride[0] \times h + m, si$$

If `padding` is non-zero, then the input is implicitly zero-padded on both sides for `padding` number of points.

FLATTEN



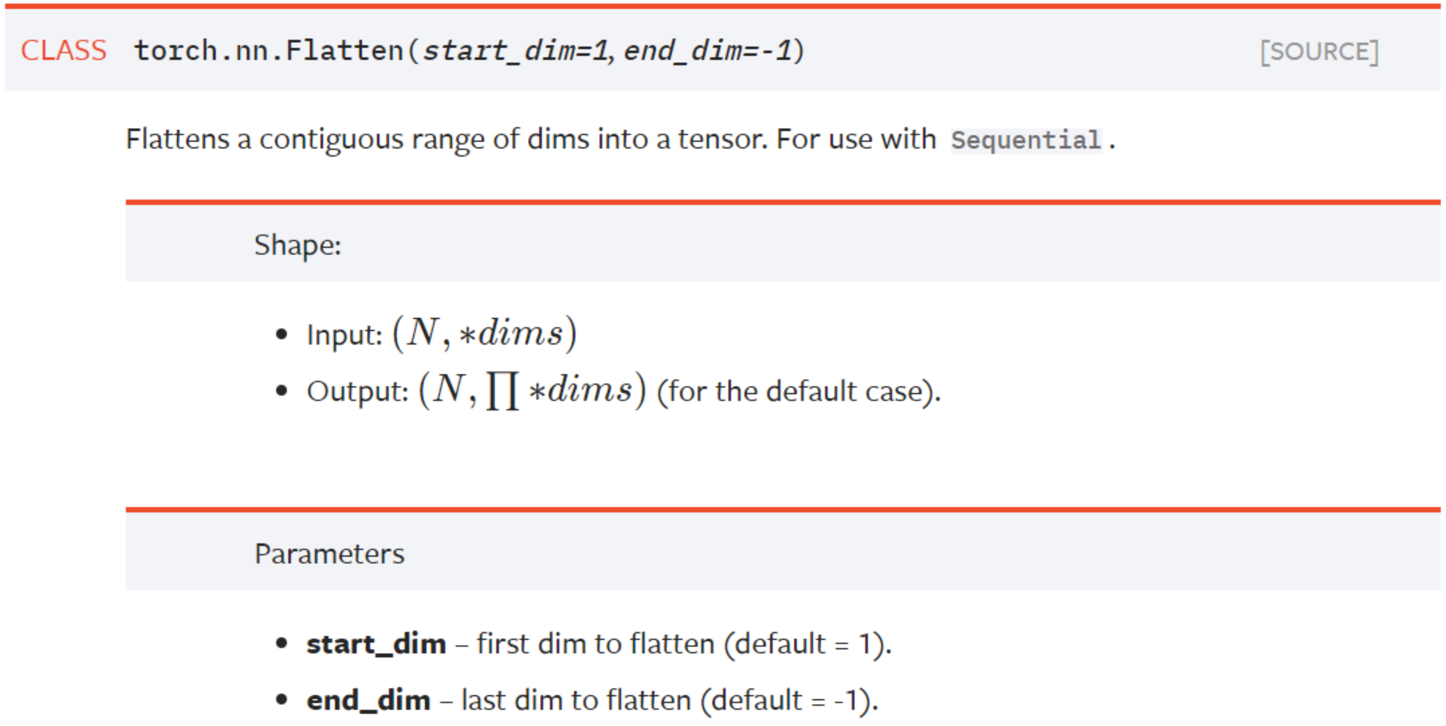
1	1	0
4	2	1
0	2	1

Pooled Feature Map

1
1
0
4
2
1
0
2
1

FLATTEN 程式碼

- torch.nn.Flatten()



CLASS `torch.nn.Flatten(start_dim=1, end_dim=-1)` [SOURCE]

Flattens a contiguous range of dims into a tensor. For use with `Sequential`.

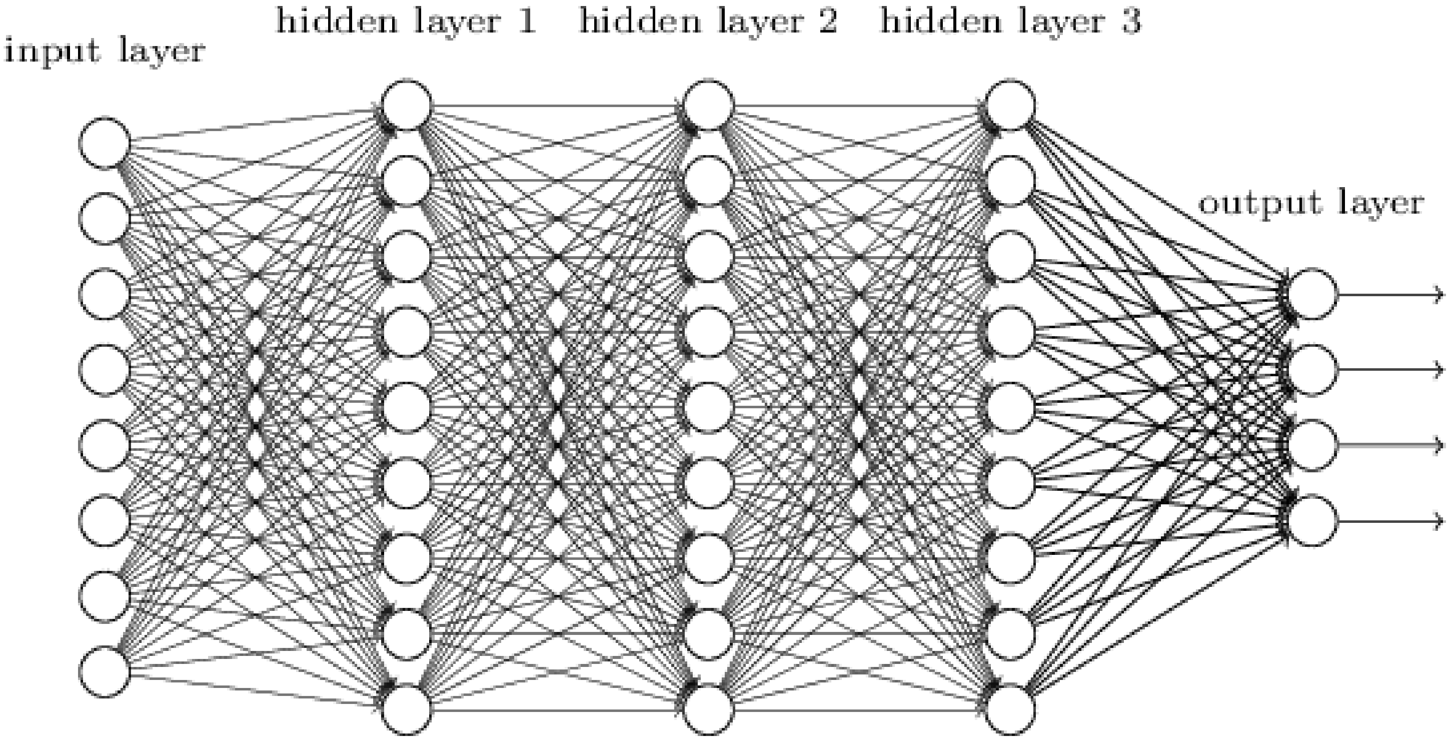
Shape:

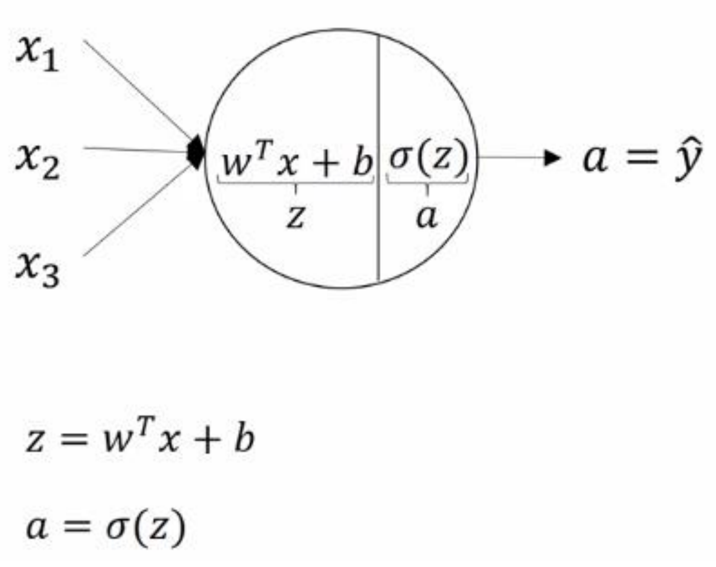
- Input: $(N, *dims)$
- Output: $(N, \prod *dims)$ (for the default case).

Parameters

- **start_dim** – first dim to flatten (default = 1).
- **end_dim** – last dim to flatten (default = -1).

FULL CONNECTED LAYER(CONT.)





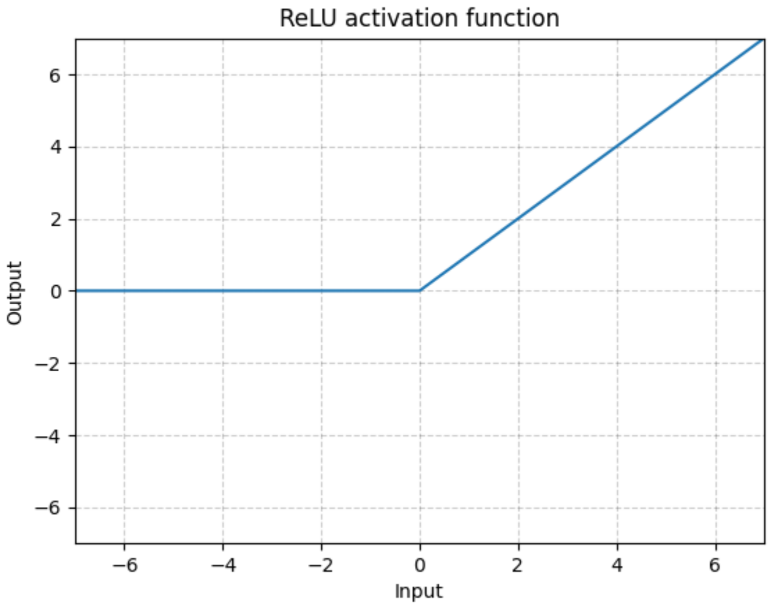
$$z = w^T x + b$$

$$a = \sigma(z)$$

ACTIVATION FUNCTION

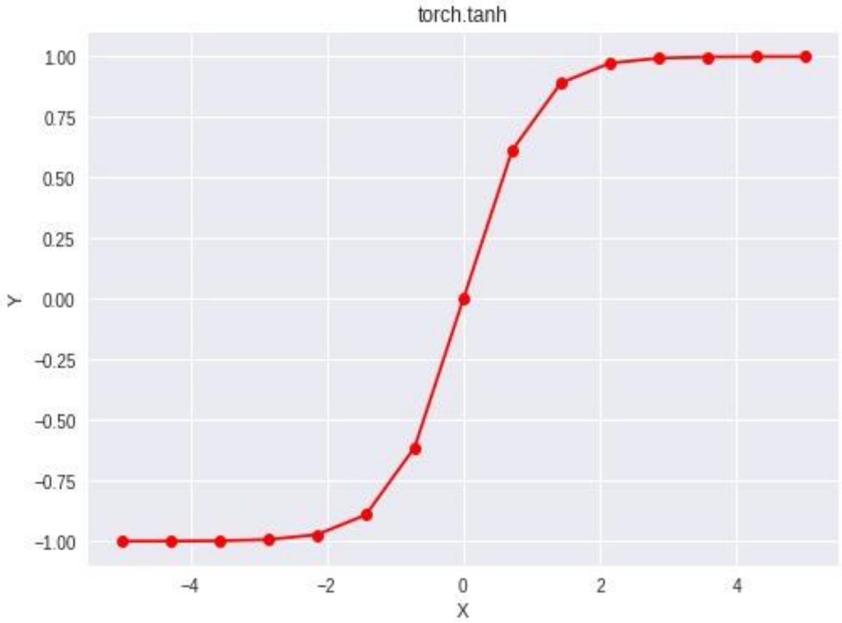
nn.ReLU()

$$\mathrm{ReLU}(x) = (x)^+ = \max(0, x)$$



nn.Tanh()

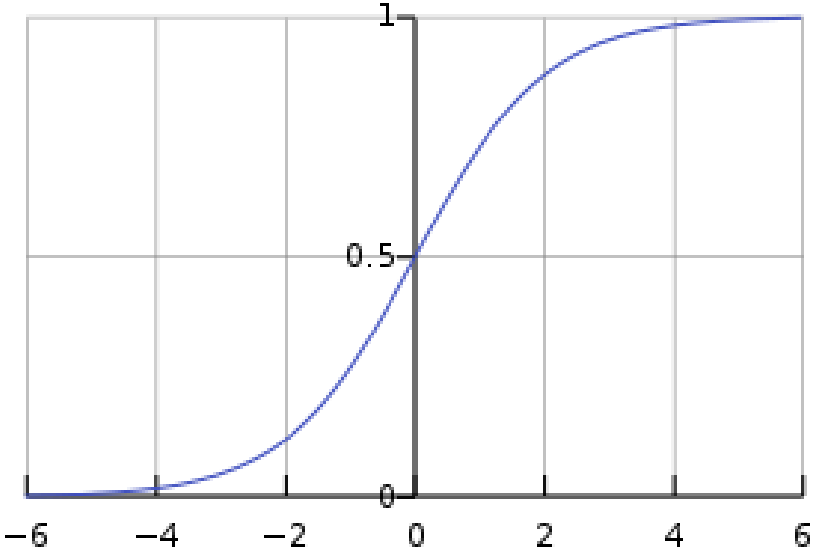
$$\mathrm{Tanh}(x) = \tanh(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$



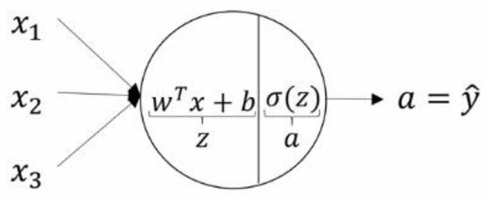
ACTIVATION FUNCTION(CONT.)

torch.sigmoid()

$$\text{Sigmoid}(x) = \sigma(x) = \frac{1}{1 + \exp(-x)}$$



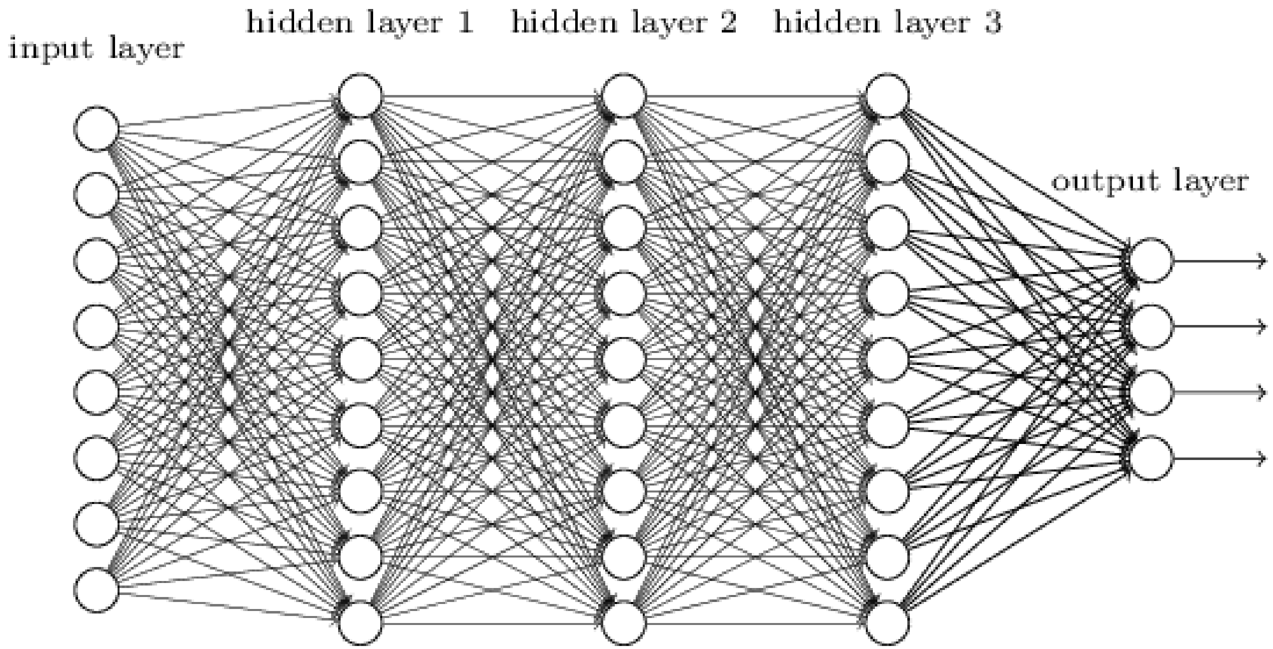
ACTIVATION FUNCTION(CONT.)

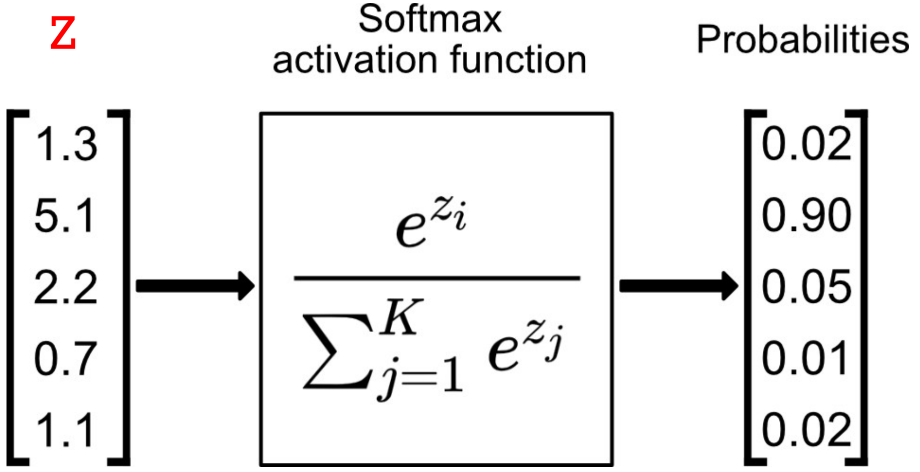


$z = w^Tx + b$

$a = \sigma(z)$

nn.softmax()





z

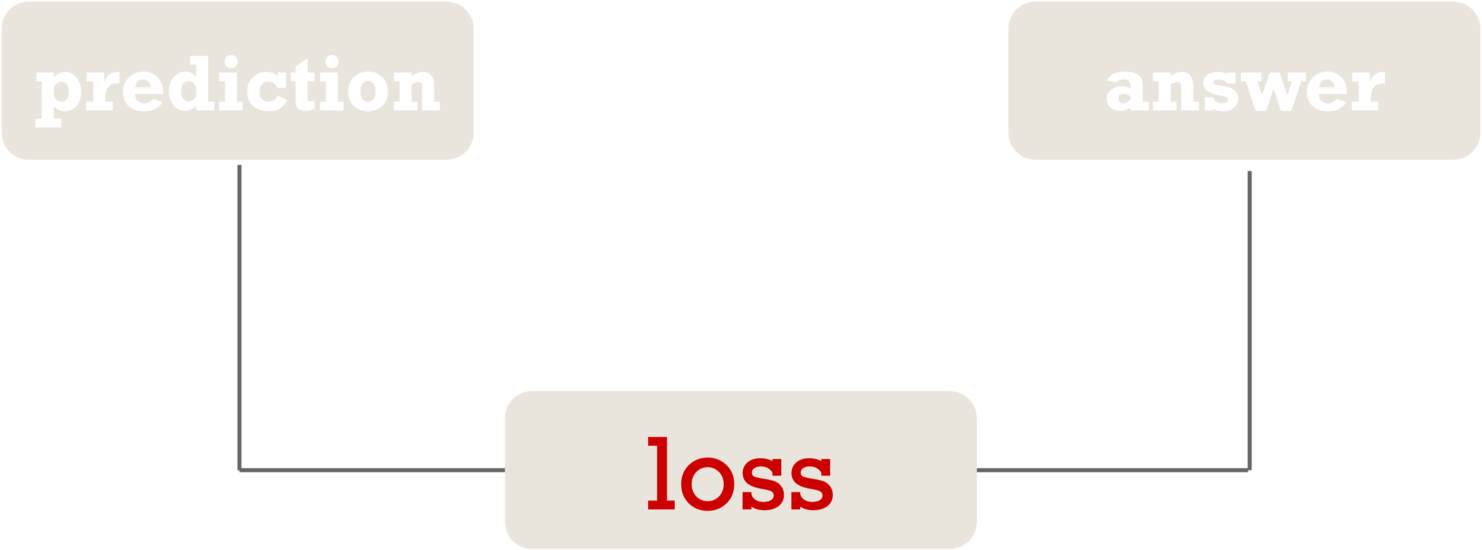
Softmax activation function

Probabilities

$$\begin{bmatrix}1.3\\5.1\\2.2\\0.7\\1.1\end{bmatrix} \rightarrow \boxed{\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}} \rightarrow \begin{bmatrix}0.02\\0.90\\0.05\\0.01\\0.02\end{bmatrix}$$

LOSS FUNCTION

- 透過計算 LOSS 值，網路能知道現在模型的狀態有多好/不好，藉由調整網路內部的權重來降低 LOSS。



LOSS FUNCTION

- 有了 LOSS 之後如何更新網路內部權重? Gradient descent

Cross Entropy (For classification)

$$H(p,q) = -\sum_{x} p(x)\,\log q(x).$$

Mean Square Error (For regression)

$$\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

OPTIMIZER

SGD

$$W \leftarrow W - \eta \frac{\partial L}{\partial W}$$

ADAM

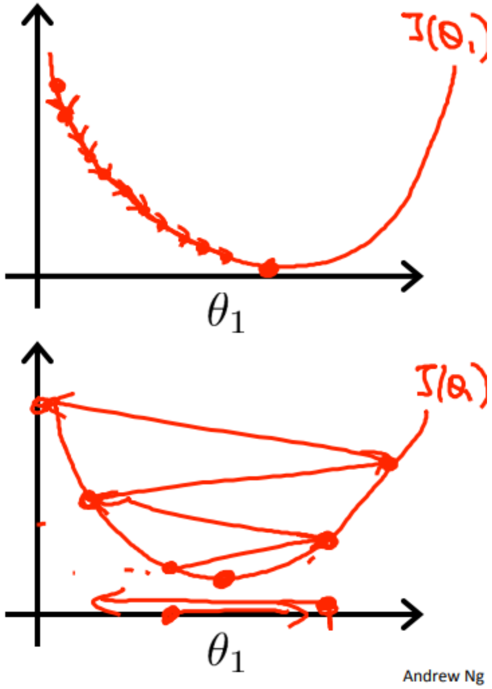
$$W \leftarrow W - \eta \frac{\widehat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

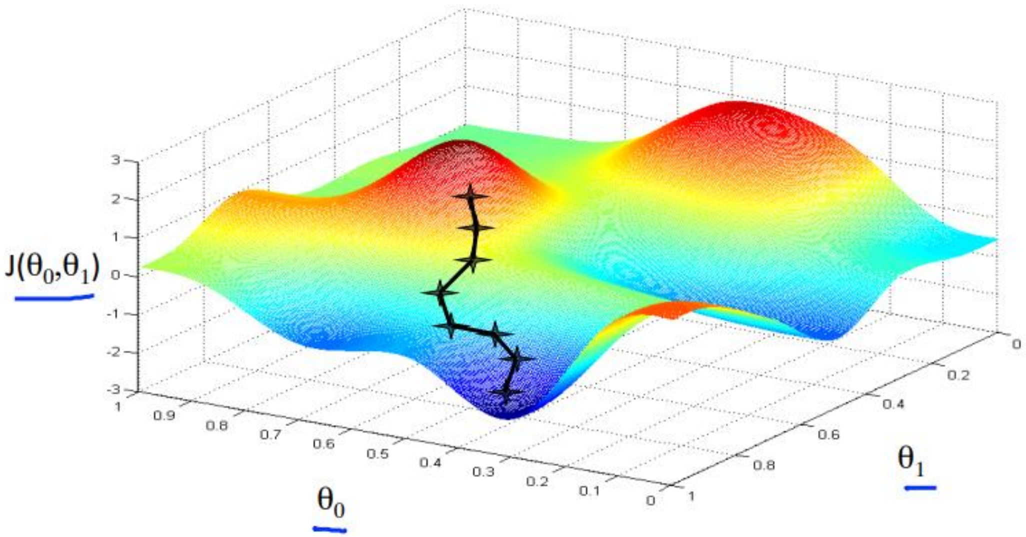
OPTIMIZER(CONT.)

$$\boxed{\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)}$$

If α is too small, gradient descent can be slow.

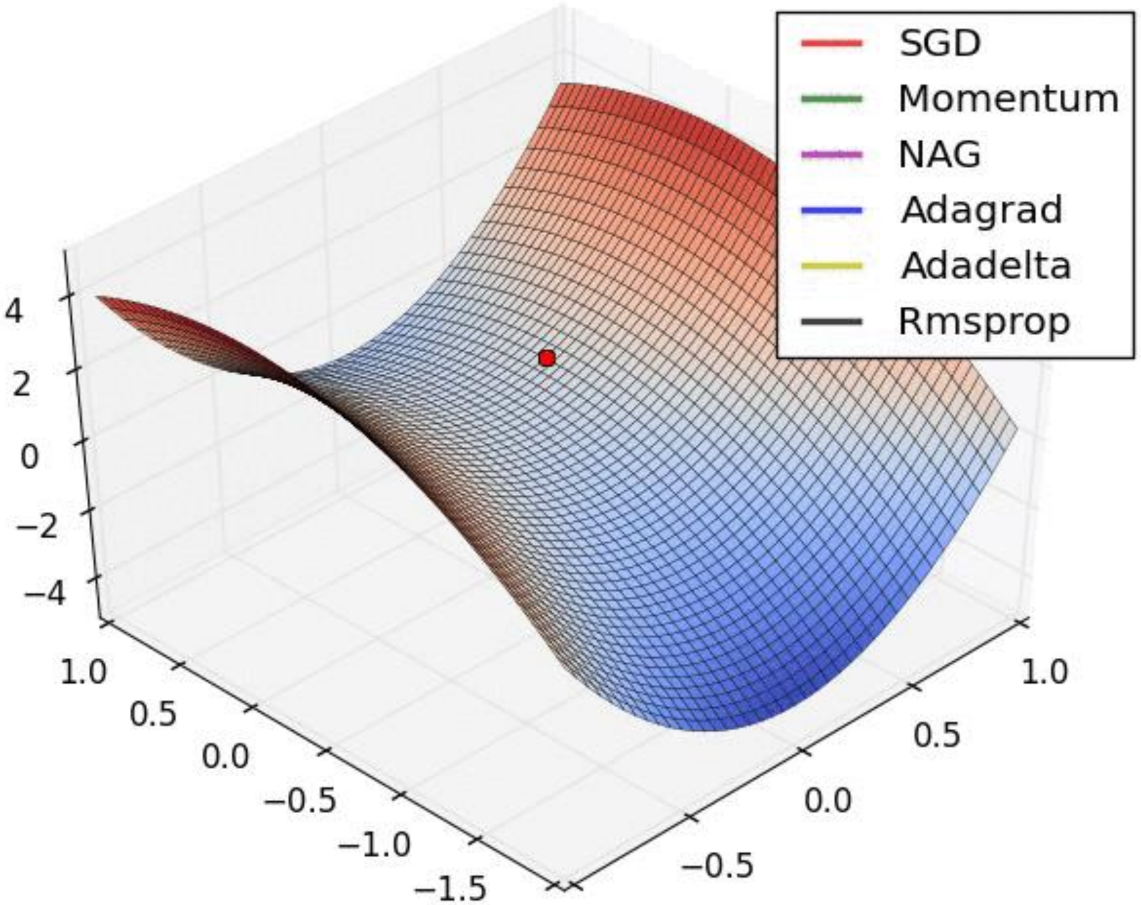
If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.





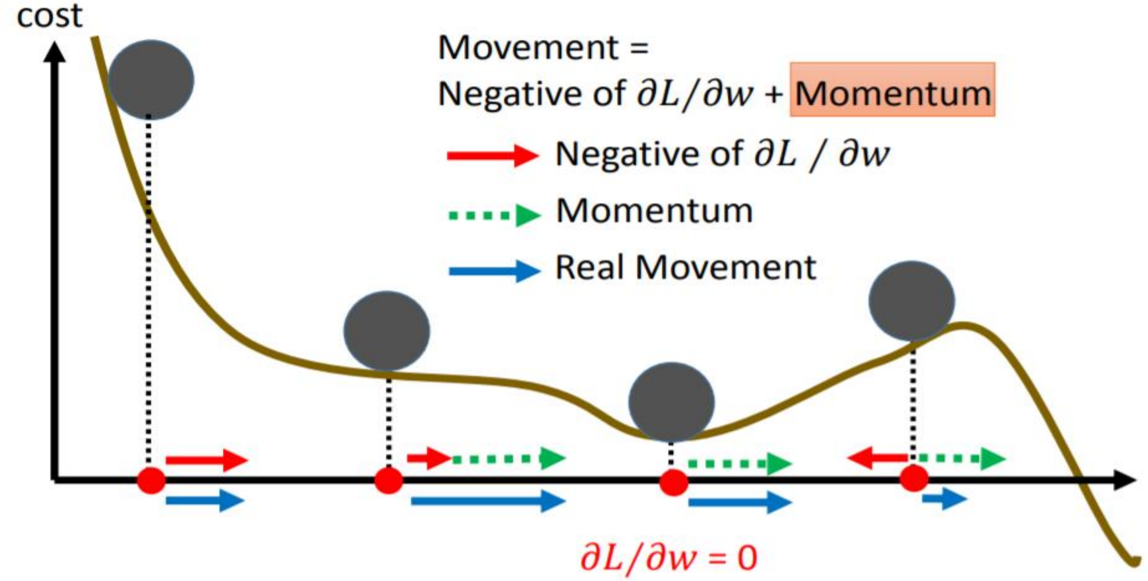
Source: Andrew Ng, Machine learning

OPTIMIZER(CONT.)



Momentum

Still not guarantee reaching global minima, but give some hope ……

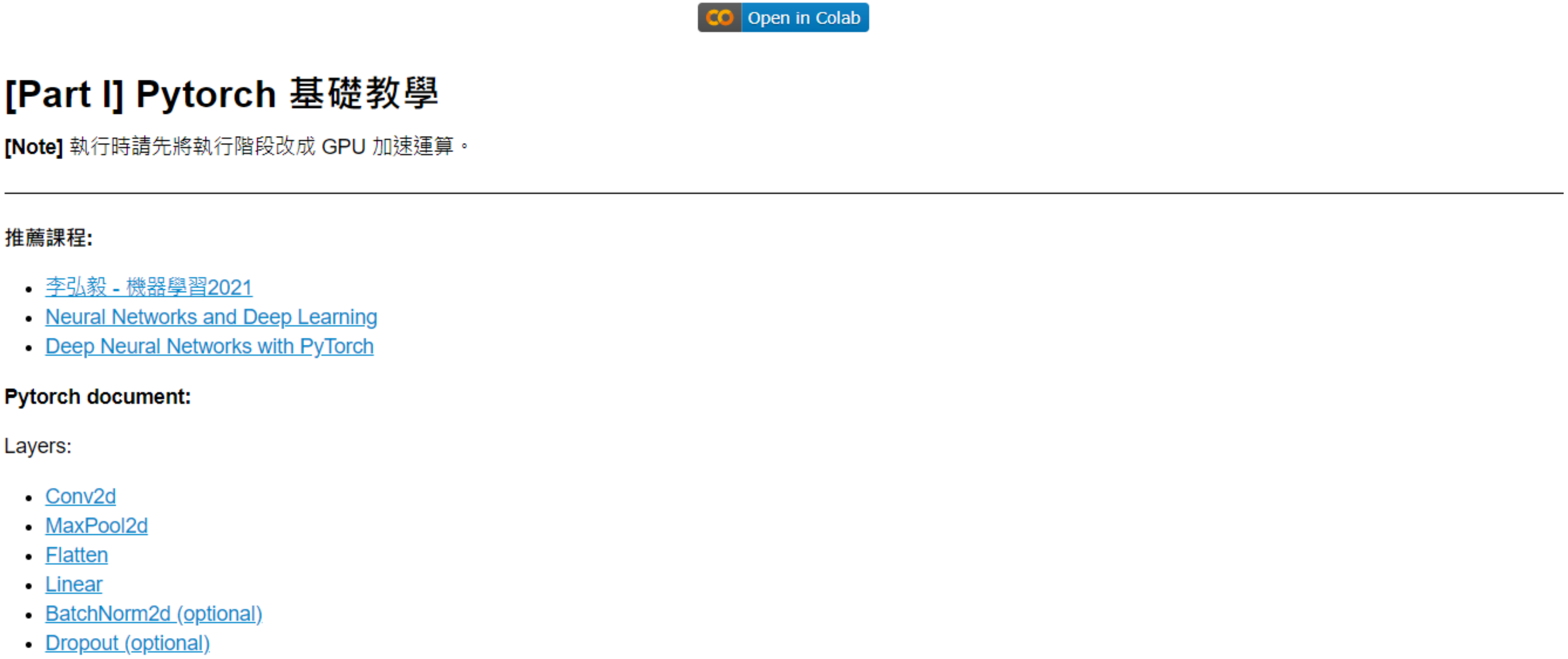


Movement = Negative of $\partial L / \partial w$ + Momentum

Source: Prof. Hung-yi Lee, 2017 Fall Machine learning

COLAB NOTEBOOK

- https://github.com/maglili/pytorch_intro/blob/main/resources/mnist_tutorial.ipynb



[Part I] Pytorch 基礎教學

[Note] 執行時請先將執行階段改成 GPU 加速運算。

推薦課程:

- 李弘毅 - 機器學習2021
- Neural Networks and Deep Learning
- Deep Neural Networks with PyTorch

Pytorch document:

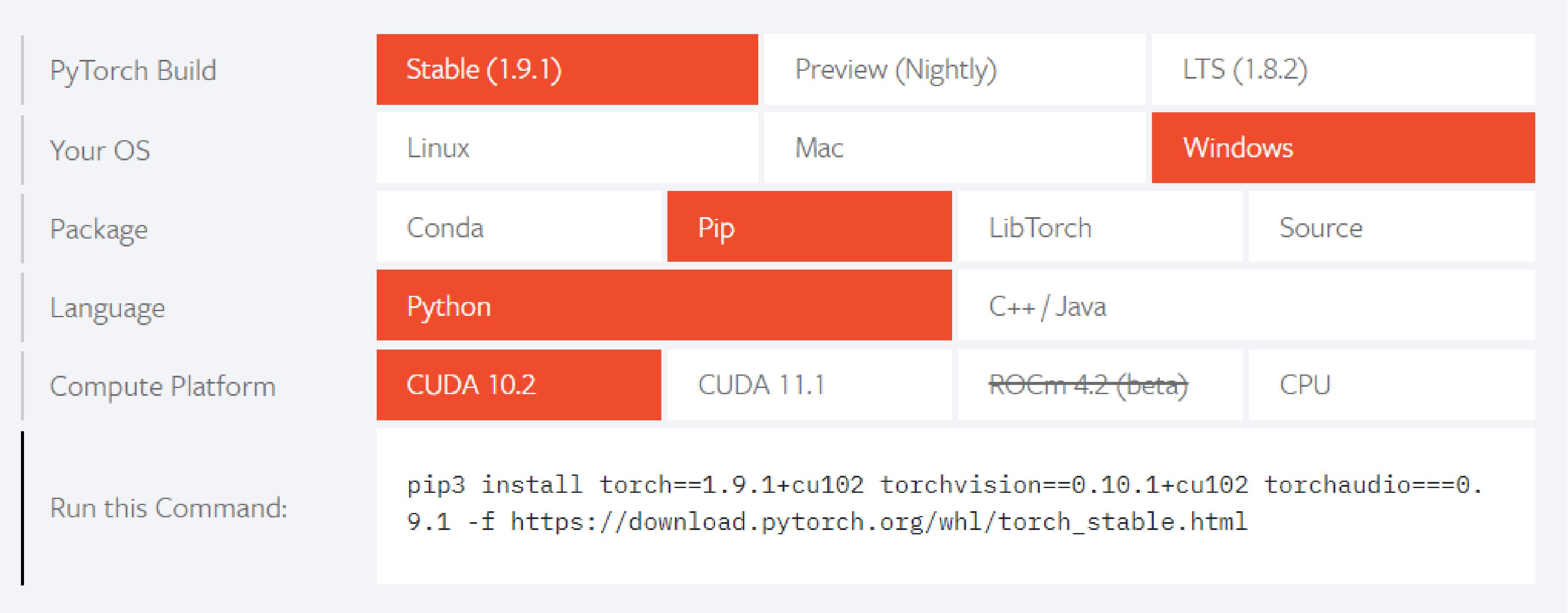
Layers:

- Conv2d
- MaxPool2d
- Flatten
- Linear
- BatchNorm2d (optional)
- Dropout (optional)

INTALL PYTORCH (GPU)

INSTALL PYTORCH

- https://pytorch.org/
- 安裝 GPU 版本需先確認 GPU 型號，以及 CUDA 版本



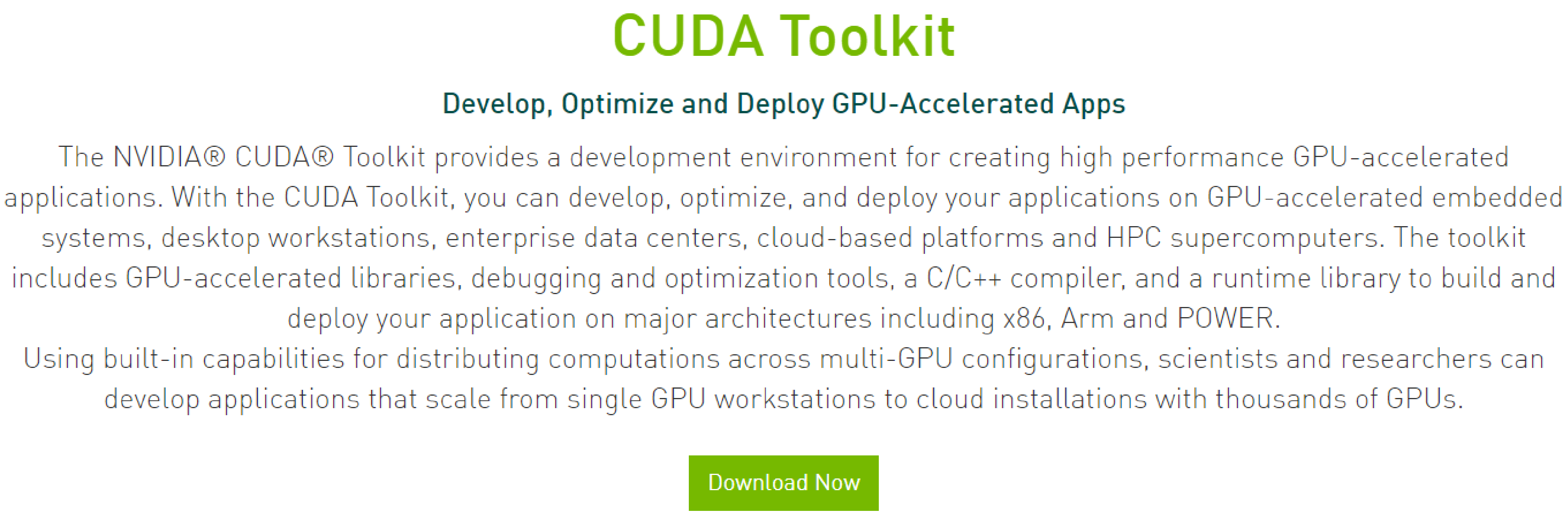
PyTorch Build	[✓ Stable (1.9.1)]	Preview (Nightly)	LTS (1.8.2)	
Your OS	Linux	Mac	[✓ Windows]	
Package	Conda	[✓ Pip]	LibTorch	Source
Language	[✓ Python]	C++ / Java		
Compute Platform	[✓ CUDA 10.2]	CUDA 11.1	~~ROCm 4.2 (beta)~~	CPU

Run this Command:

```
pip3 install torch==1.9.1+cu102 torchvision==0.10.1+cu102 torchaudio===0.
9.1 -f https://download.pytorch.org/whl/torch_stable.html
```

CUDA

- https://developer.nvidia.com/cuda-toolkit
- 需註冊 nvidia 會員才可安裝



CUDA Toolkit

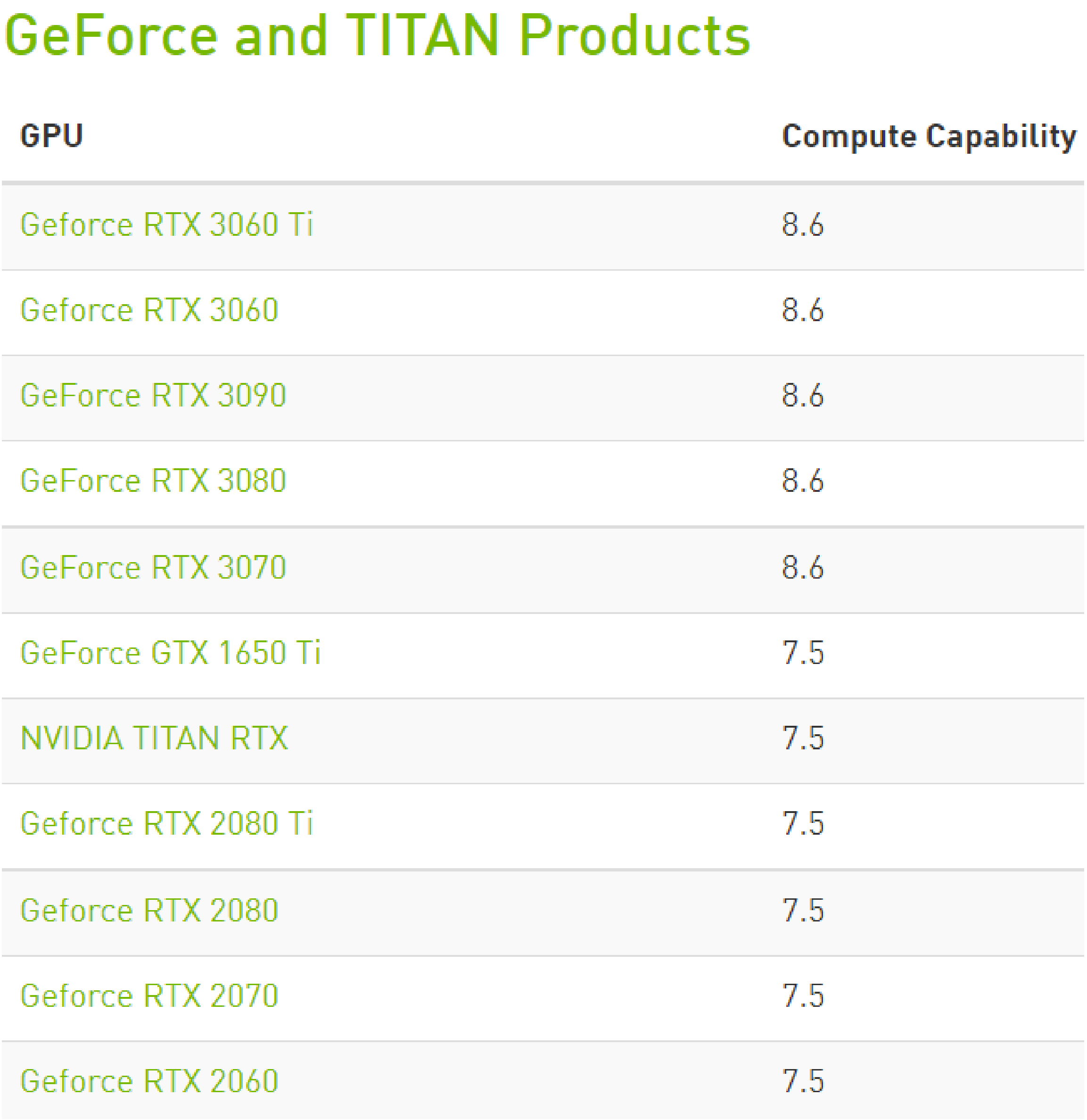
Develop, Optimize and Deploy GPU-Accelerated Apps

The NVIDIA® CUDA® Toolkit provides a development environment for creating high performance GPU-accelerated applications. With the CUDA Toolkit, you can develop, optimize, and deploy your applications on GPU-accelerated embedded systems, desktop workstations, enterprise data centers, cloud-based platforms and HPC supercomputers. The toolkit includes GPU-accelerated libraries, debugging and optimization tools, a C/C++ compiler, and a runtime library to build and deploy your application on major architectures including x86, Arm and POWER.

Using built-in capabilities for distributing computations across multi-GPU configurations, scientists and researchers can develop applications that scale from single GPU workstations to cloud installations with thousands of GPUs.

CUDA(CONT.)

- https://developer.nvidia.com/cuda-gpus
- 確認 GPU 是否支援 CUDA



GeForce and TITAN Products

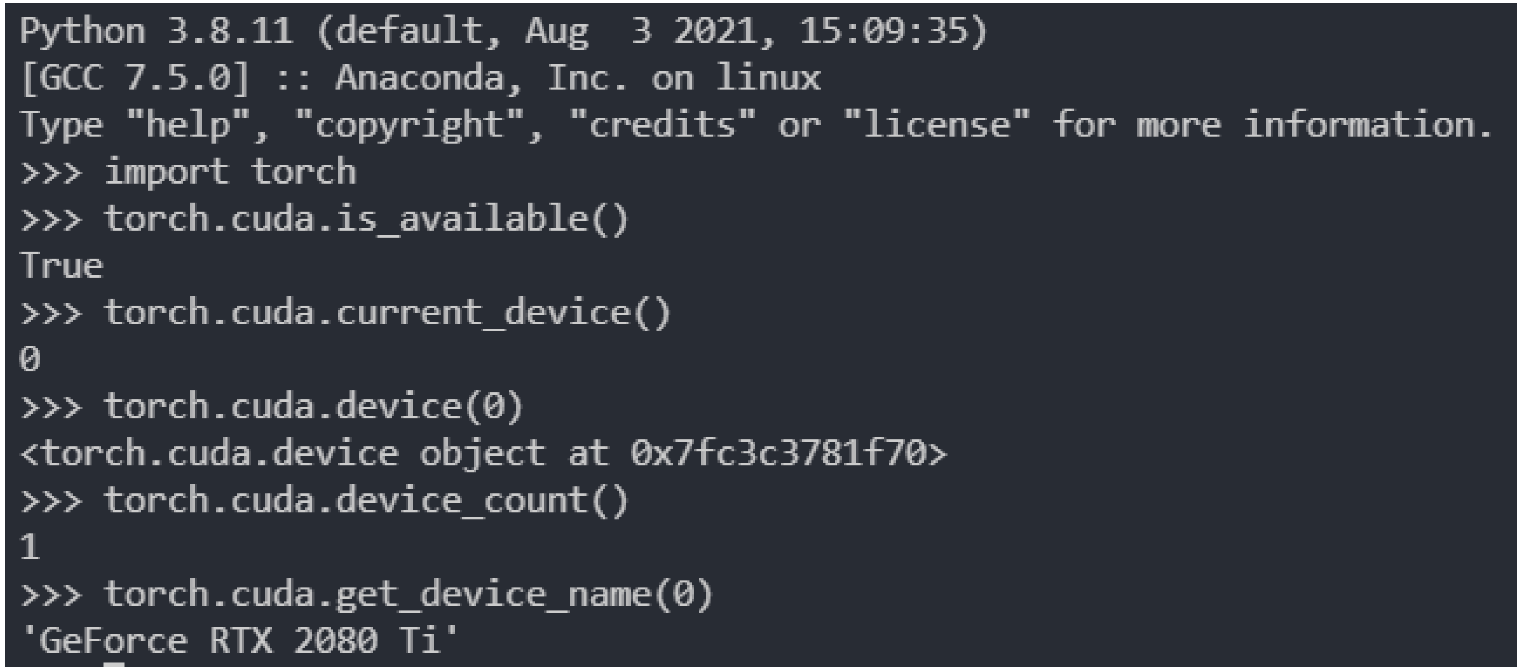
GPU	Compute Capability
Geforce RTX 3060 Ti	8.6
Geforce RTX 3060	8.6
GeForce RTX 3090	8.6
GeForce RTX 3080	8.6
GeForce RTX 3070	8.6
GeForce GTX 1650 Ti	7.5
NVIDIA TITAN RTX	7.5
Geforce RTX 2080 Ti	7.5
Geforce RTX 2080	7.5
Geforce RTX 2070	7.5
Geforce RTX 2060	7.5

CUDA(CONT.)

- Installing Tensorflow with CUDA, cuDNN and GPU support on Windows 10
- 1. 安裝指定版本的 CUDA
- 2. 將 CUDA 路徑加入環境變數
- 3. 按照 CUDA 版本安裝 cuDNN

CUDA(CONT.)

- 依照 CUDA 版本安裝 GPU 版本的 pytorch
- 檢查 GPU 版本之 PYTORCH 安裝是否正常?

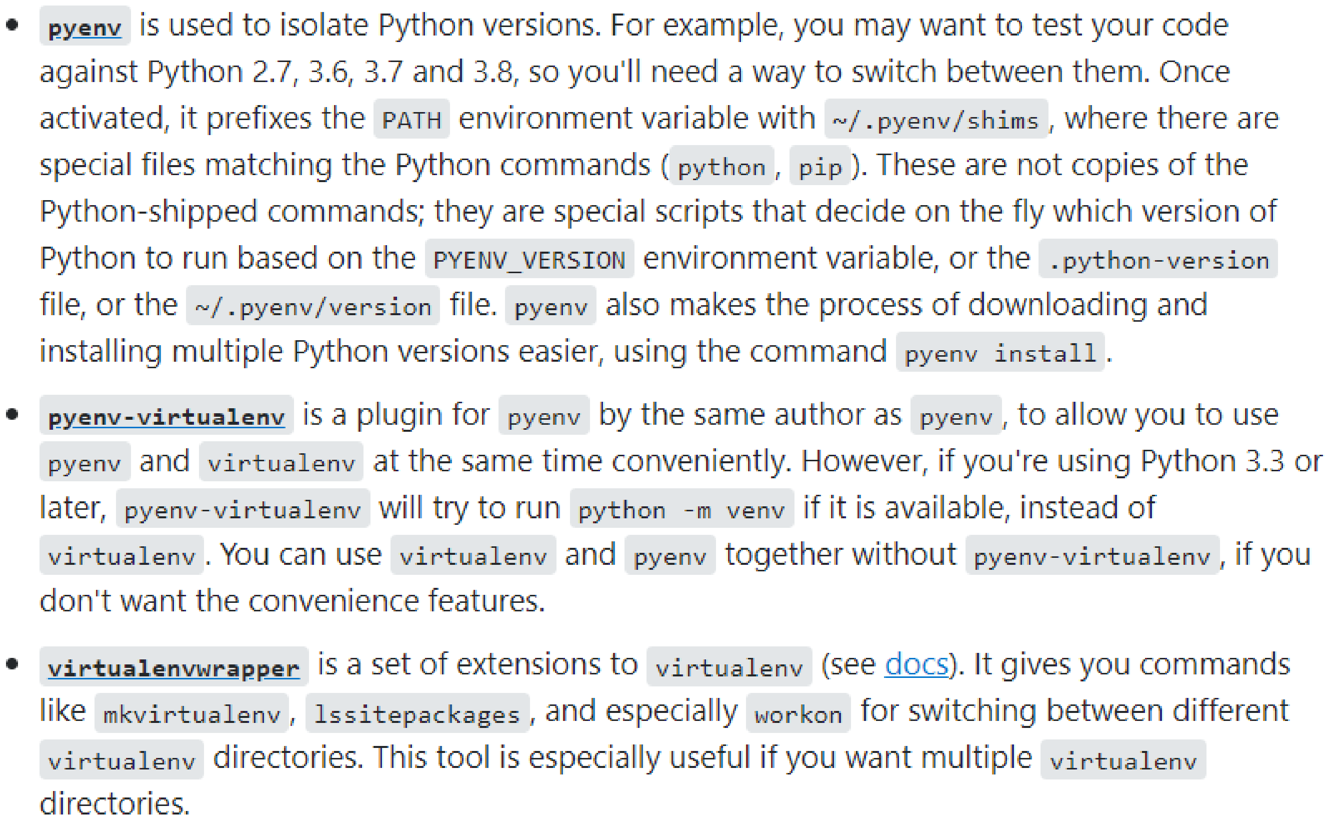


```
Python 3.8.11 (default, Aug  3 2021, 15:09:35)
[GCC 7.5.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>> torch.cuda.current_device()
0
>>> torch.cuda.device(0)
<torch.cuda.device object at 0x7fc3c3781f70>
>>> torch.cuda.device_count()
1
>>> torch.cuda.get_device_name(0)
'GeForce RTX 2080 Ti'
```

cosbi7

VIRTUAL ENV

- What is the difference between venv, pyvenv, pyenv, virtualenv, virtualenvwrapper, pipenv, etc?
- 管理虛擬環境
  - Venv
  - Virtualenvwrapper
  - conda

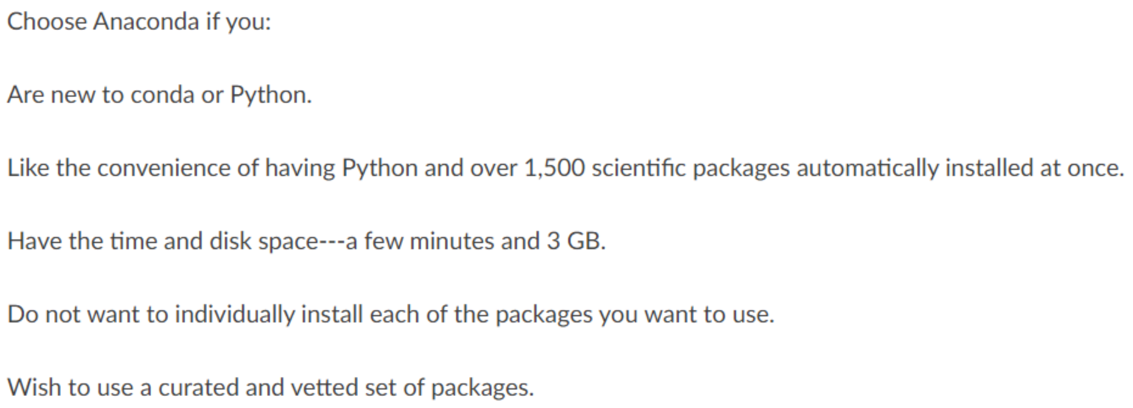


- `pyenv` is used to isolate Python versions. For example, you may want to test your code against Python 2.7, 3.6, 3.7 and 3.8, so you'll need a way to switch between them. Once activated, it prefixes the `PATH` environment variable with `~/.pyenv/shims`, where there are special files matching the Python commands (`python`, `pip`). These are not copies of the Python-shipped commands; they are special scripts that decide on the fly which version of Python to run based on the `PYENV_VERSION` environment variable, or the `.python-version` file, or the `~/.pyenv/version` file. `pyenv` also makes the process of downloading and installing multiple Python versions easier, using the command `pyenv install`.
- `pyenv-virtualenv` is a plugin for `pyenv` by the same author as `pyenv`, to allow you to use `pyenv` and `virtualenv` at the same time conveniently. However, if you're using Python 3.3 or later, `pyenv-virtualenv` will try to run `python -m venv` if it is available, instead of `virtualenv`. You can use `virtualenv` and `pyenv` together without `pyenv-virtualenv`, if you don't want the convenience features.
- `virtualenvwrapper` is a set of extensions to `virtualenv` (see docs). It gives you commands like `mkvirtualenv`, `lssitepackages`, and especially `workon` for switching between different `virtualenv` directories. This tool is especially useful if you want multiple `virtualenv` directories.

CONDA

Anaconda or Miniconda?

Anaconda



Choose Anaconda if you:

Are new to conda or Python.

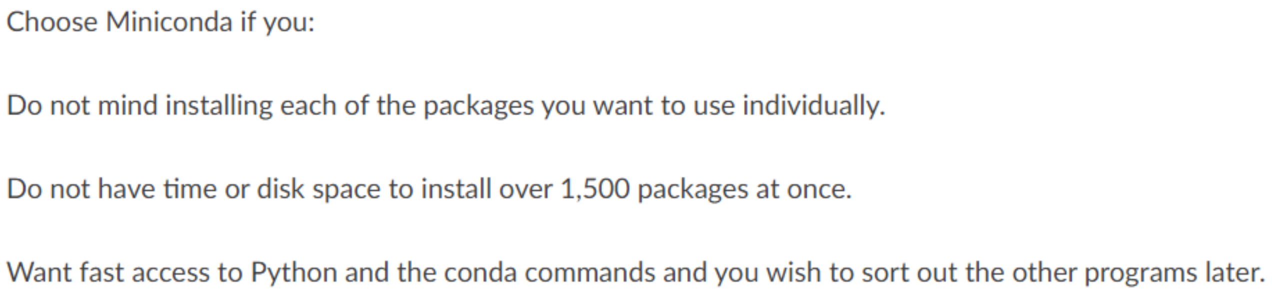
Like the convenience of having Python and over 1,500 scientific packages automatically installed at once.

Have the time and disk space---a few minutes and 3 GB.

Do not want to individually install each of the packages you want to use.

Wish to use a curated and vetted set of packages.

Mini-conda



Choose Miniconda if you:

Do not mind installing each of the packages you want to use individually.

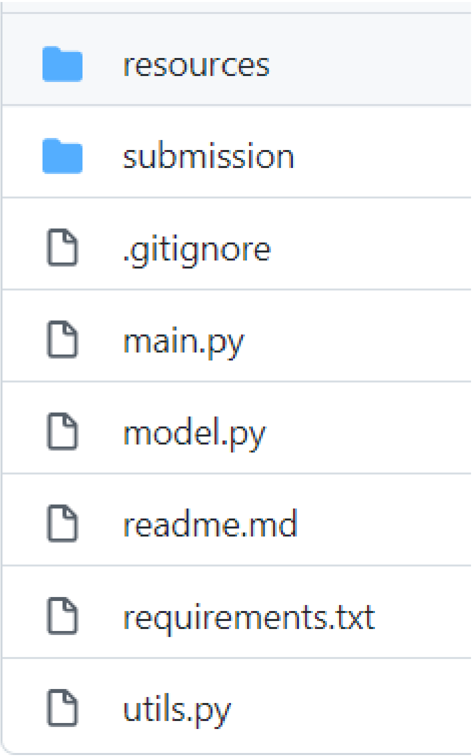
Do not have time or disk space to install over 1,500 packages at once.

Want fast access to Python and the conda commands and you wish to sort out the other programs later.

DL PROJECT STRCTURE

DL PROJECT

- https://github.com/maglili/pytorch_intro



resources

submission

.gitignore

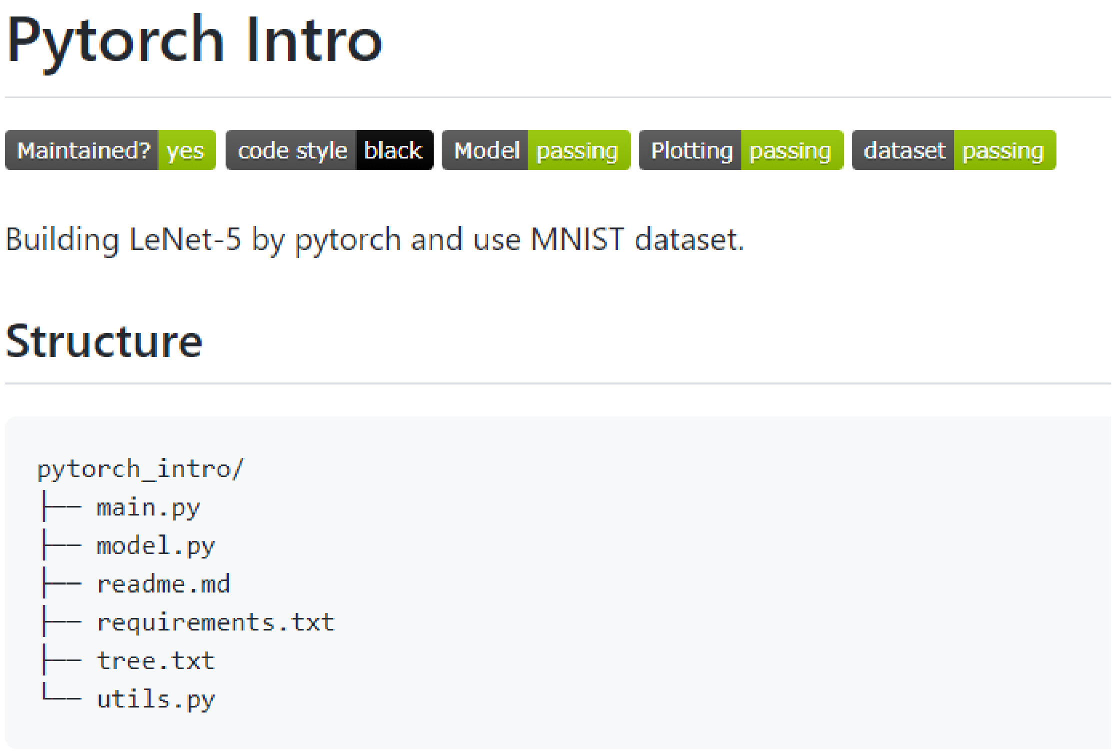
main.py

model.py

readme.md

requirements.txt

utils.py



Pytorch Intro

Maintained? yes | code style black | Model passing | Plotting passing | dataset passing

Building LeNet-5 by pytorch and use MNIST dataset.

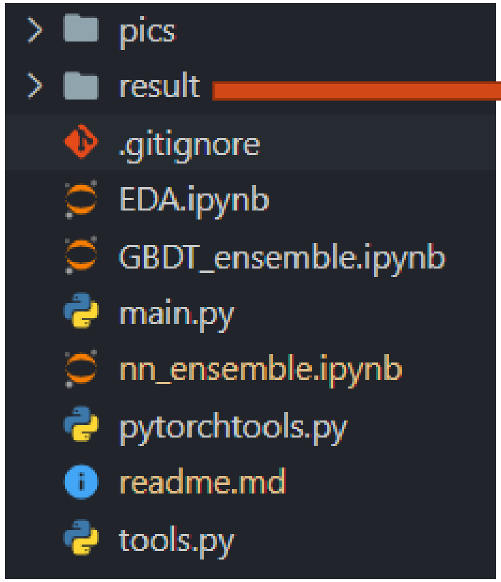
Structure

```
pytorch_intro/
├── main.py
├── model.py
├── readme.md
├── requirements.txt
├── tree.txt
└── utils.py
```

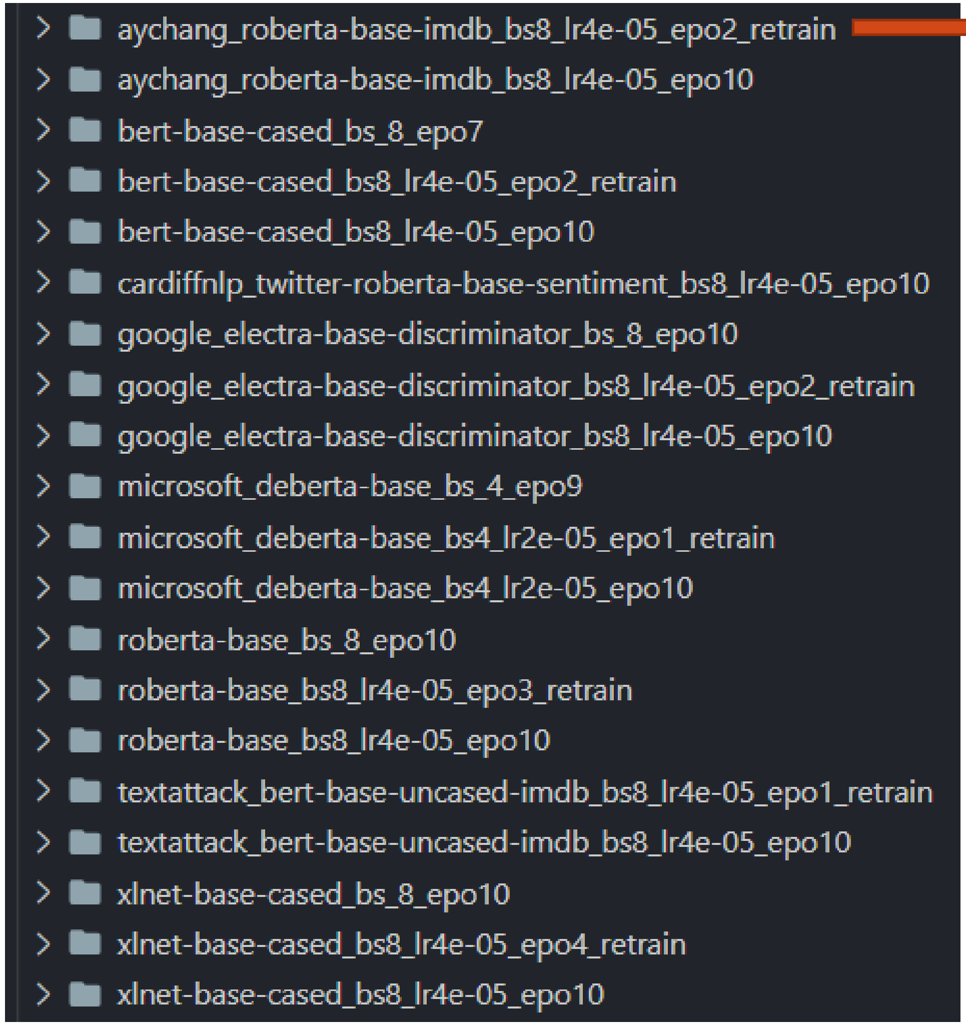
STRCTURE

事先規劃好整個 repo 的階層架構

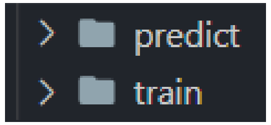
e.g. 以不同資料夾存放不同的實驗，資料夾名稱以模型跟參數來命名，如下圖



```
> pics
> result
  .gitignore
  EDA.ipynb
  GBDT_ensemble.ipynb
  main.py
  nn_ensemble.ipynb
  pytorchtools.py
  readme.md
  tools.py
```



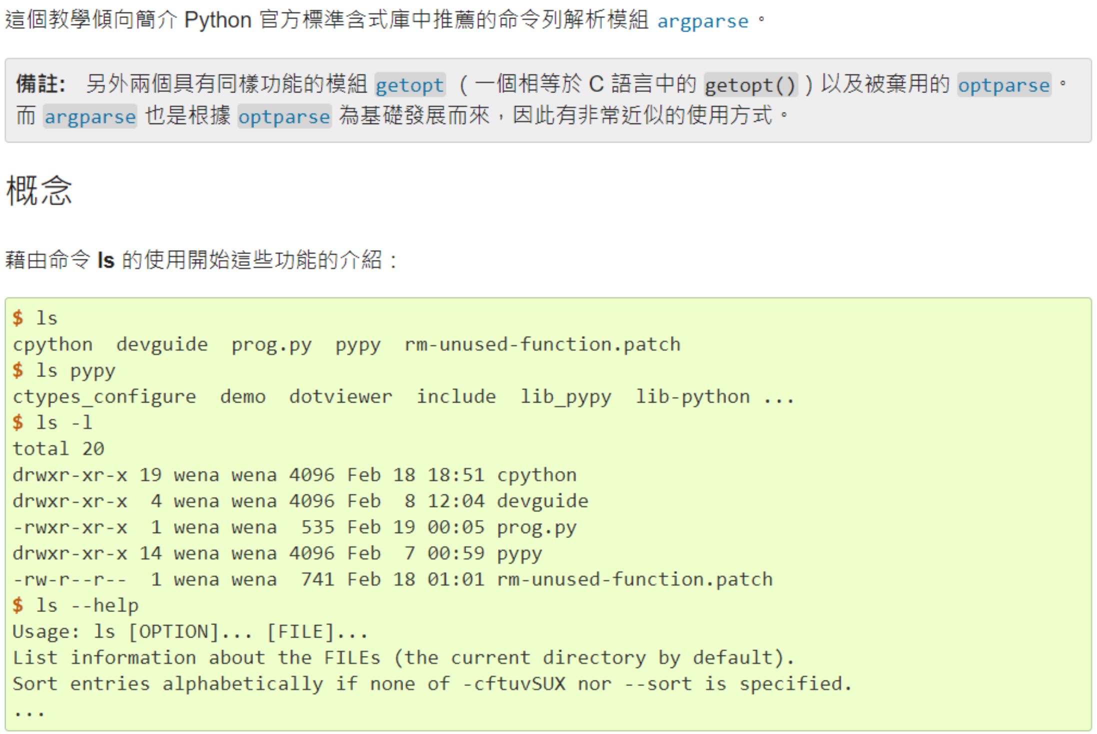
```
> aychang_roberta-base-imdb_bs8_lr4e-05_epo2_retrain
> aychang_roberta-base-imdb_bs8_lr4e-05_epo10
> bert-base-cased_bs_8_epo7
> bert-base-cased_bs8_lr4e-05_epo2_retrain
> bert-base-cased_bs8_lr4e-05_epo10
> cardiffnlp_twitter-roberta-base-sentiment_bs8_lr4e-05_epo10
> google_electra-base-discriminator_bs_8_epo10
> google_electra-base-discriminator_bs8_lr4e-05_epo2_retrain
> google_electra-base-discriminator_bs8_lr4e-05_epo10
> microsoft_deberta-base_bs_4_epo9
> microsoft_deberta-base_bs4_lr2e-05_epo1_retrain
> microsoft_deberta-base_bs4_lr2e-05_epo10
> roberta-base_bs_8_epo10
> roberta-base_bs8_lr4e-05_epo3_retrain
> roberta-base_bs8_lr4e-05_epo10
> textattack_bert-base-uncased-imdb_bs8_lr4e-05_epo1_retrain
> textattack_bert-base-uncased-imdb_bs8_lr4e-05_epo10
> xlnet-base-cased_bs_8_epo10
> xlnet-base-cased_bs8_lr4e-05_epo4_retrain
> xlnet-base-cased_bs8_lr4e-05_epo10
```



```
> predict
> train
```

ARGPARSE

- [Argparse 教學— Python 3.9.7 說明文件]
- [Python argparse 教學：比 sys.argv 更好用，讓命令列引數整潔又有序]



這個教學傾向簡介 Python 官方標準含式庫中推薦的命令列解析模組 argparse 。

備註: 另外兩個具有同樣功能的模組 getopt （一個相等於 C 語言中的 getopt() ）以及被棄用的 optparse 。 而 argparse 也是根據 optparse 為基礎發展而來，因此有非常近似的使用方式。

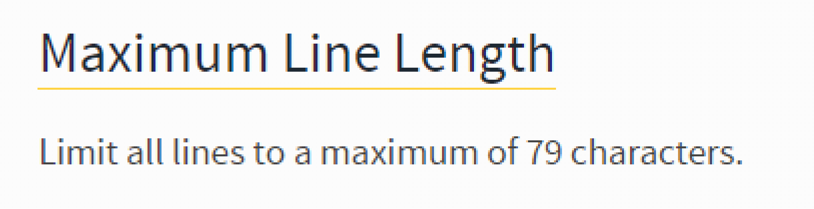
概念

藉由命令 **ls** 的使用開始這些功能的介紹：

```
$ ls
cpython  devguide  prog.py  pypy  rm-unused-function.patch
$ ls pypy
ctypes_configure  demo  dotviewer  include  lib_pypy  lib-python ...
$ ls -l
total 20
drwxr-xr-x 19 wena wena 4096 Feb 18 18:51 cpython
drwxr-xr-x  4 wena wena 4096 Feb  8 12:04 devguide
-rwxr-xr-x  1 wena wena  535 Feb 19 00:05 prog.py
drwxr-xr-x 14 wena wena 4096 Feb  7 00:59 pypy
-rw-r--r--  1 wena wena  741 Feb 18 01:01 rm-unused-function.patch
$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
...
```

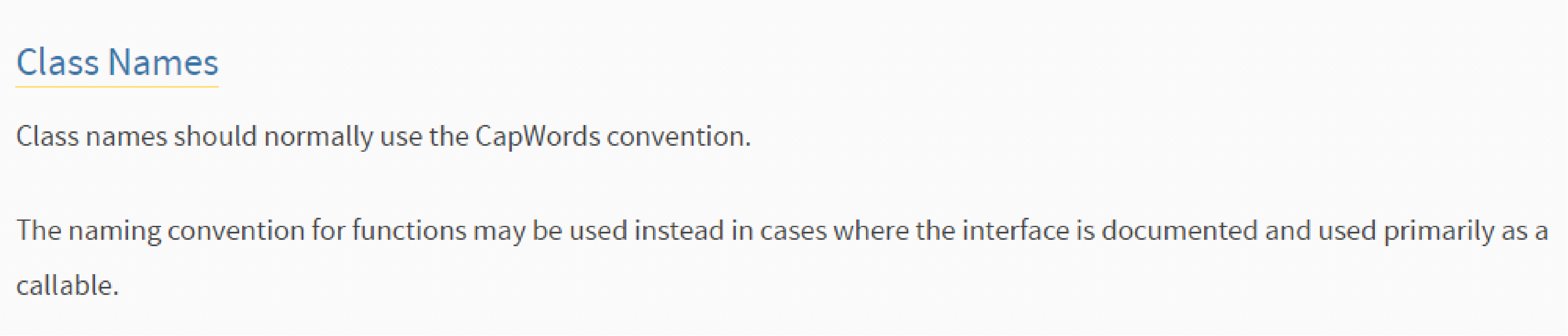
PEP8

- PEP 0 -- Index of Python Enhancement Proposals (PEPs)
- PEP 8 -- Style Guide for Python Code



Maximum Line Length

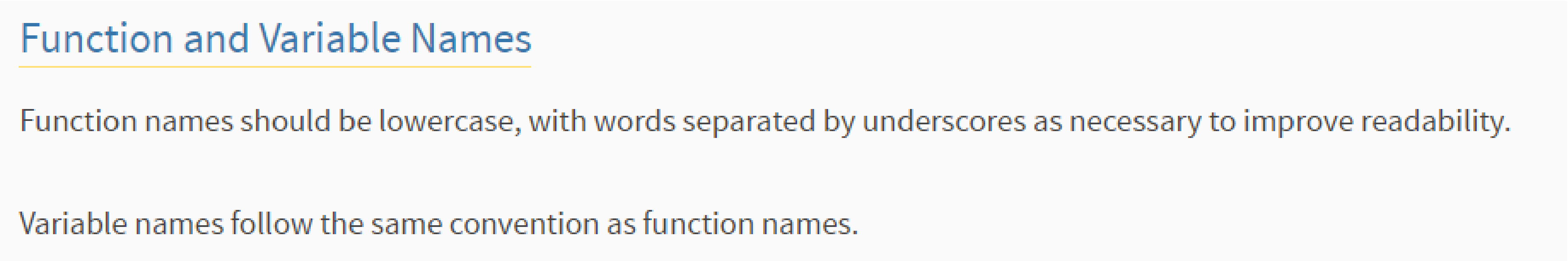
Limit all lines to a maximum of 79 characters.



Class Names

Class names should normally use the CapWords convention.

The naming convention for functions may be used instead in cases where the interface is documented and used primarily as a callable.



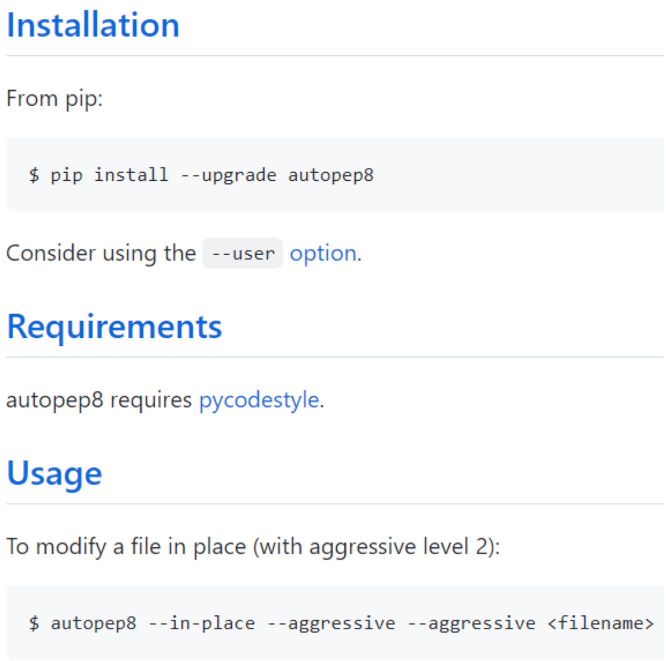
Function and Variable Names

Function names should be lowercase, with words separated by underscores as necessary to improve readability.

Variable names follow the same convention as function names.

FORMATTER

Autopep8



Installation

From pip:

```
$ pip install --upgrade autopep8
```

Consider using the `--user` option.

Requirements

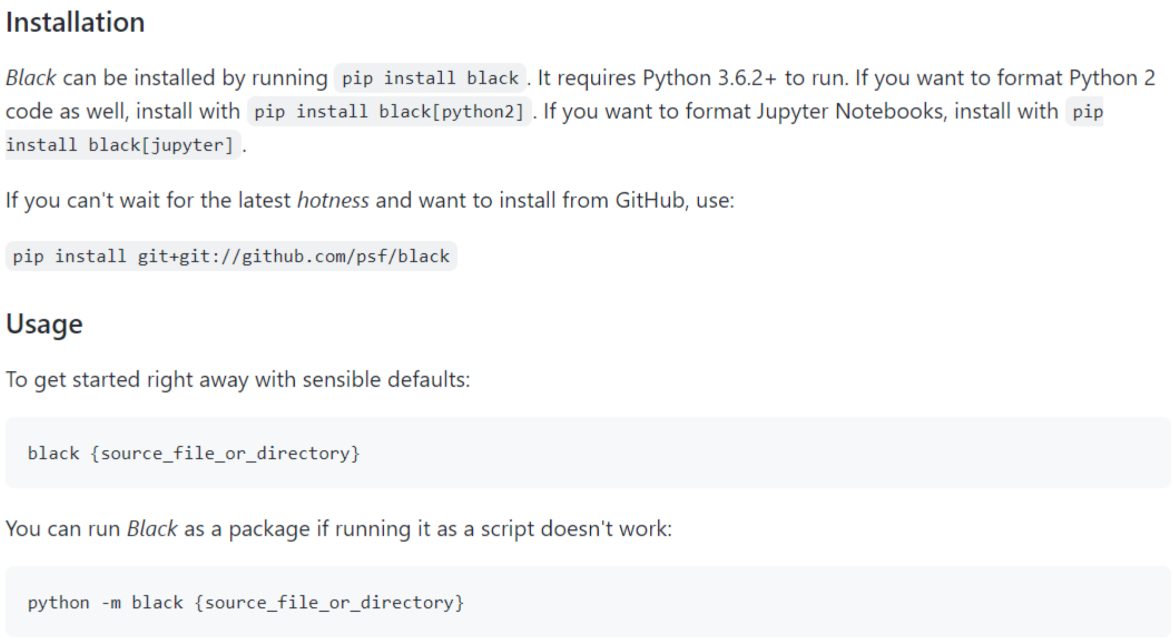
autopep8 requires pycodestyle.

Usage

To modify a file in place (with aggressive level 2):

```
$ autopep8 --in-place --aggressive --aggressive <filename>
```

Black



Installation

Black can be installed by running `pip install black`. It requires Python 3.6.2+ to run. If you want to format Python 2 code as well, install with `pip install black[python2]`. If you want to format Jupyter Notebooks, install with `pip install black[jupyter]`.

If you can't wait for the latest *hotness* and want to install from GitHub, use:

`pip install git+git://github.com/psf/black`

Usage

To get started right away with sensible defaults:

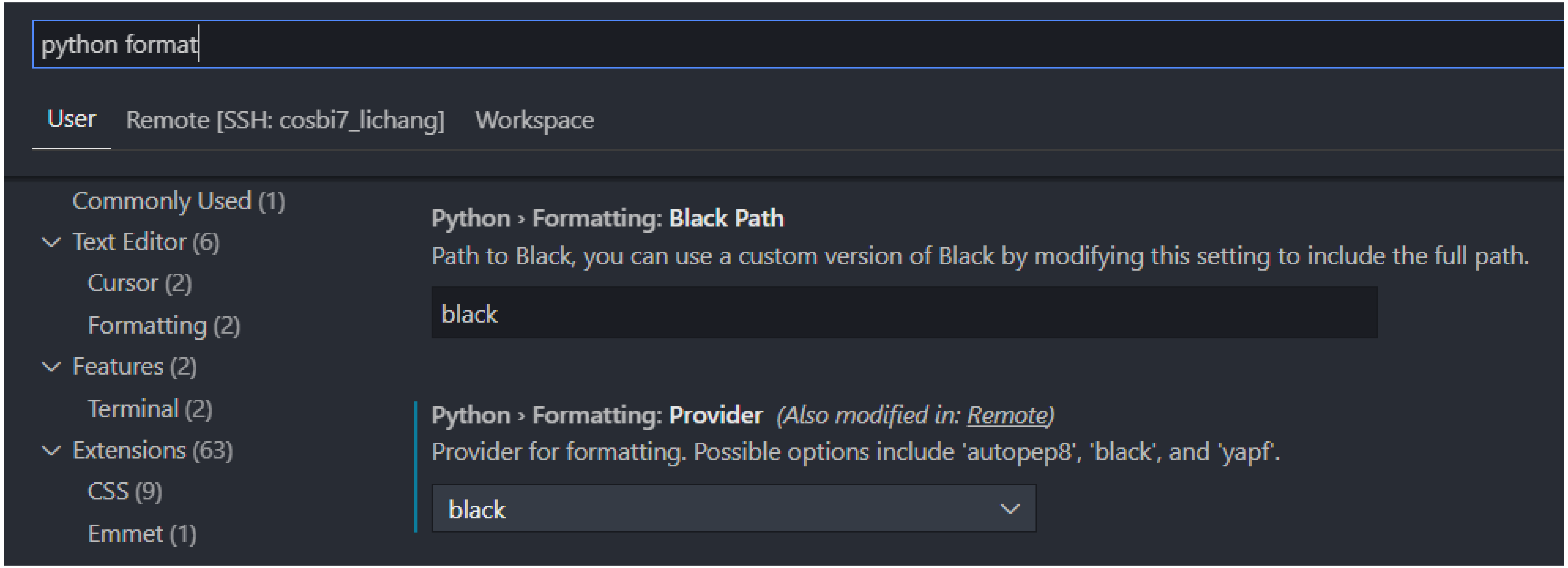
```
black {source_file_or_directory}
```

You can run *Black* as a package if running it as a script doesn't work:

```
python -m black {source_file_or_directory}
```

VSCODE

- https://dev.to/adamlombard/how-to-use-the-black-python-code-formatter-in-vscode-3lo0



python format

Python › Formatting: Black Path

Path to Black, you can use a custom version of Black by modifying this setting to include the full path.

black

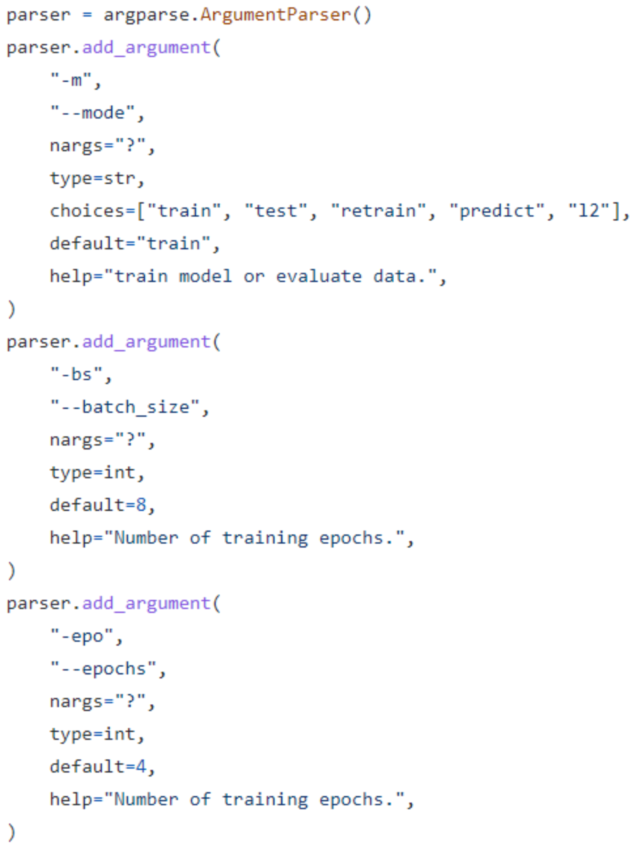
Python › Formatting: Provider *(Also modified in: Remote)*

Provider for formatting. Possible options include 'autopep8', 'black', and 'yapf'.

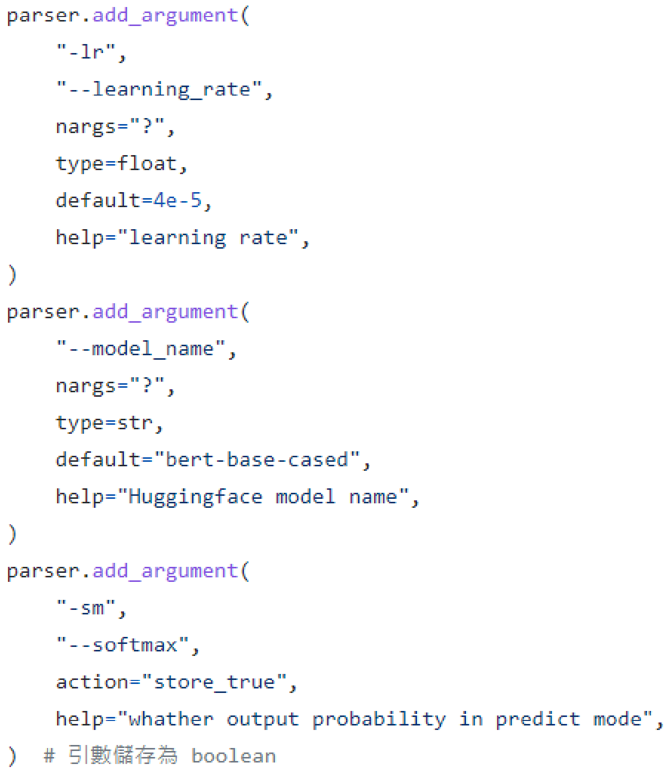
black

REFERENCE

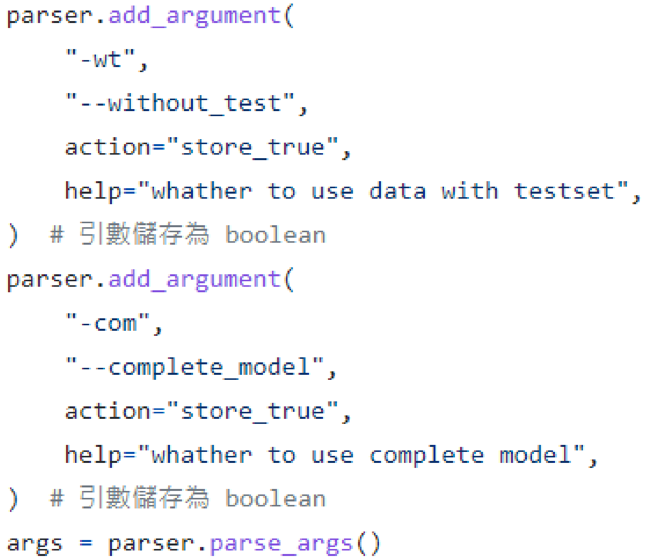
- https://github.com/maglili/ai_cup-movie_comment



```
parser = argparse.ArgumentParser()
parser.add_argument(
    "-m",
    "--mode",
    nargs="?",
    type=str,
    choices=["train", "test", "retrain", "predict", "l2"],
    default="train",
    help="train model or evaluate data.",
)
parser.add_argument(
    "-bs",
    "--batch_size",
    nargs="?",
    type=int,
    default=8,
    help="Number of training epochs.",
)
parser.add_argument(
    "-epo",
    "--epochs",
    nargs="?",
    type=int,
    default=4,
    help="Number of training epochs.",
)
```



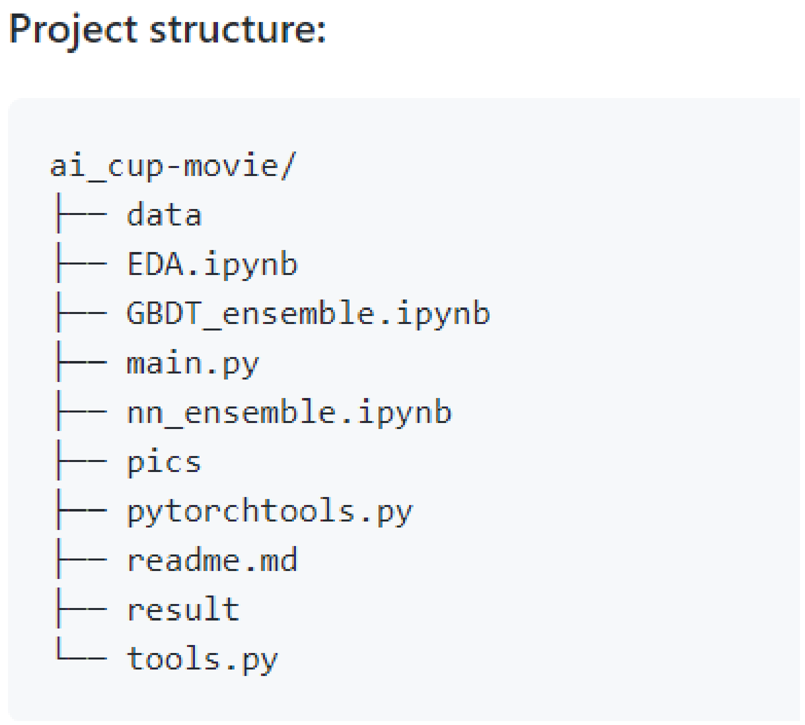
```
parser.add_argument(
    "-lr",
    "--learning_rate",
    nargs="?",
    type=float,
    default=4e-5,
    help="learning rate",
)
parser.add_argument(
    "--model_name",
    nargs="?",
    type=str,
    default="bert-base-cased",
    help="Huggingface model name",
)
parser.add_argument(
    "-sm",
    "--softmax",
    action="store_true",
    help="whather output probability in predict mode",
)  # 引數儲存為 boolean
```



```
parser.add_argument(
    "-wt",
    "--without_test",
    action="store_true",
    help="whather to use data with testset",
)  # 引數儲存為 boolean
parser.add_argument(
    "-com",
    "--complete_model",
    action="store_true",
    help="whather to use complete model",
)  # 引數儲存為 boolean
args = parser.parse_args()
```

REFERENCE(CONT.)

- https://github.com/maglili/dqn-pytorch



Project structure:

```
ai_cup-movie/
├── data
├── EDA.ipynb
├── GBDT_ensemble.ipynb
├── main.py
├── nn_ensemble.ipynb
├── pics
├── pytorchtools.py
├── readme.md
├── result
└── tools.py
```

THANK YOU FOR YOUR ATTENTION