Data generator in Python for Keras (& Tensorflow)

Sean Yu, 2017/08/17

What is data generator and why we need it?

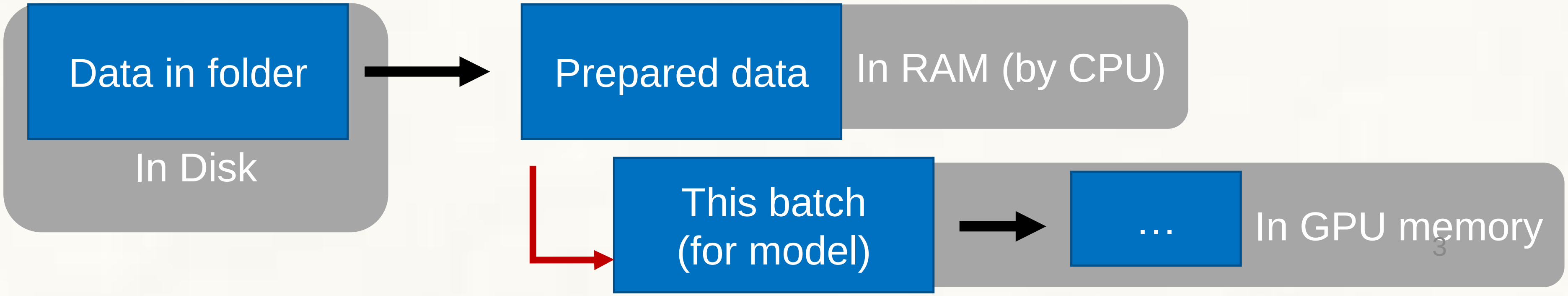
- 假設大家都寫過 keras or tensorflow
  - Generally, we will have …
    - training set / validation set / testing set
  - We always load them ALL into memory
    - MNIST – 60k images, 28 x 28 x 1
    - Cifar10 – 60k images, 32 x 32 x 3
    - EASY!

- However, if you have 100k+ 400 x 300 x 3 images, it is impossible to load them all.
  - We need to real-time load data

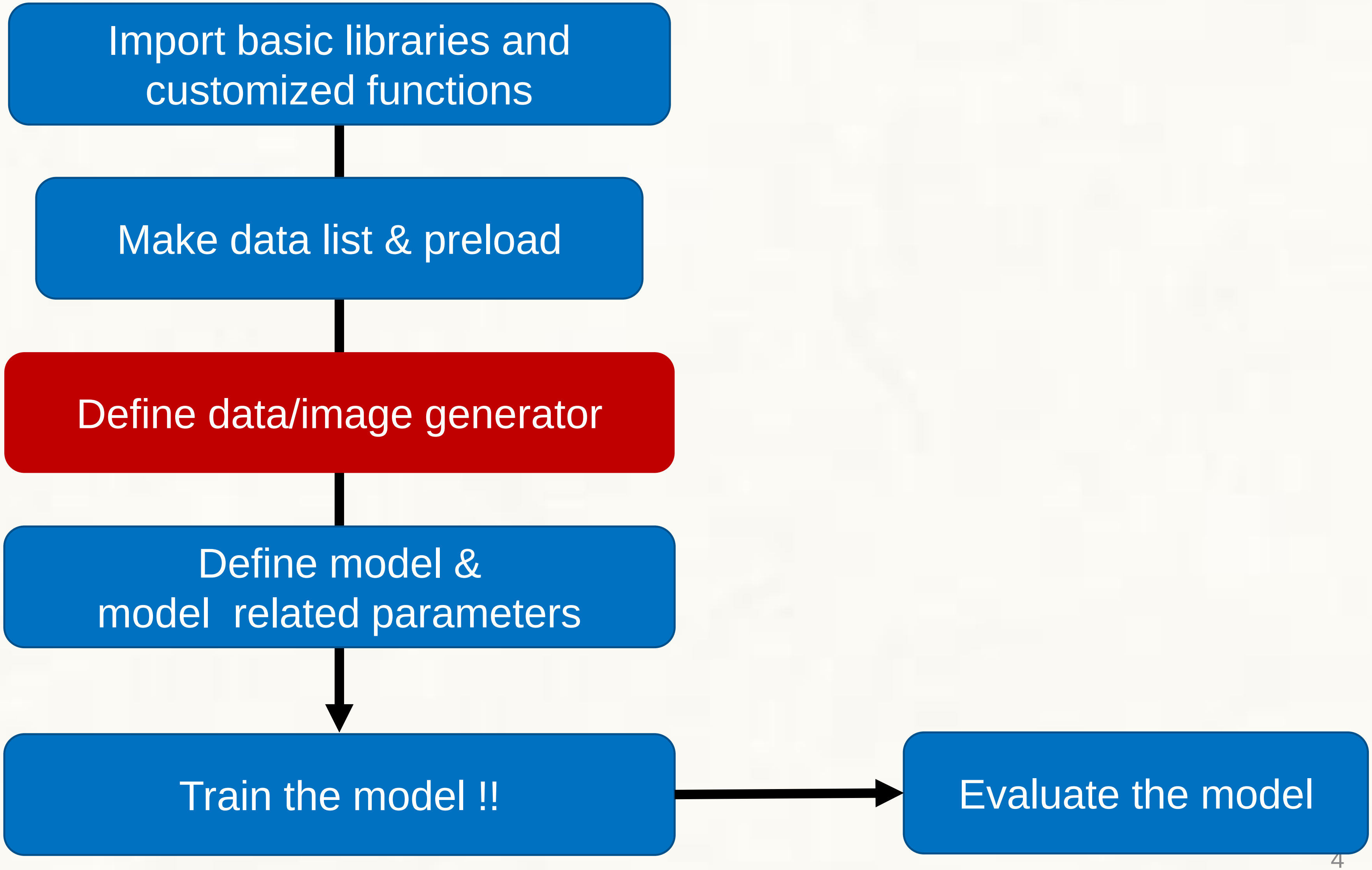
Real-time data loading

- With real-time data loading, we can do many manipulations on the data.
    - Augmentation
    - Add random noise
    - …
    - 你想對 data 幹什麼就幹什麼

- Principle of data generator
    - A infinite loop that can **'yield'** data when being requested



Coding flow



Import basic libraries and customized functions

Make data list & preload

Define data/image generator

Define model & model related parameters

Train the model !!

Evaluate the model

Today’s example data

- Kaggle, cats and dogs classification
  - https://www.kaggle.com/c/dogs-vs-cats
- Keras blog 上面其實有類似的 example code 了, 改寫一下而已
- Classification problem: cat or dog
  - Training set: 25k
  - Testing set 12.5k

Note

- Train generator 跟 Keras 的 Image generator 在 yield 打到怎麼辦?
    - 看 code, 簡單來說, break 它

Note

- Validation augmentation?
    - We only need to augment it at begin (not dynamically!)
    - 乾五郝? – 我的經驗, 好像有用捏
    - 增加 validation 的難度 (複雜性) – 避免 validation 進步太神速而太早被 earlystop

Note

- Validation augmentation?
  - We only need to augment it at begin (not dynamically!)
  - 乾五郝? – 我的經驗, 好像有用捏
  - 增加 validation 的難度 (複雜性) – 避免 validation 進步太神速而太早被 earlystop

Note

- Testing set 可以做 augmentation 嗎?

Note

- Testing set 可以做 augmentation 嗎?
  - 本質上不是不同影像

Conclusion

Data generator 很好用
我希望人人都有一個

其他 GPU 相關使用注意事項

- 在單張卡上, 限制每個 GPU memory fraction
- 在多張卡的機器上 (如 server), 選定使用特定編號之 GPU
- 關閉 jupyter notebook 占用之 GPU 空間

其他 GPU 相關使用注意事項

- 在單張卡上, 限制每個 GPU memory fraction

```
import tensorflow as tf
from keras.backend.tensorflow_backend import set_session
config = tf.ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.5 # take 50% of gpu memory
set_session(tf.Session(config=config))
```

- 在多張卡的機器上 (如 server), 選定使用特定編號之 GPU

- 關閉 jupyter notebook 占用之 GPU 空間

其他 GPU 相關使用注意事項

- 在單張卡上, 限制每個 GPU memory fraction
- 在多張卡的機器上 (如 server), 選定使用特定編號之 GPU

- For Unix (python script)

In the terminal,

```
CUDA_VISIBLE_DEVICES=0 python your_script.py
```

- For python script

At the script begin

```
Import os
os.environ[‘CUDA_VISIBLE_DEVICES’] = 0
```

- For jupyter notebook

At the notebook begin

```
%env CUDA_VISIBLE_DEVICES=0
```

- 關閉 jupyter notebook 占用之 GPU 空間

其他 GPU 相關使用注意事項

- 在單張卡上, 限制每個 GPU memory fraction
- 在多張卡的機器上 (如 server), 選定使用特定編號之 GPU

把這段加在 notebook 最後並執行

```
%%javascript
Jupyter.notebook.session.delete();
```

- 關閉 jupyter notebook 占用之 GPU 空間