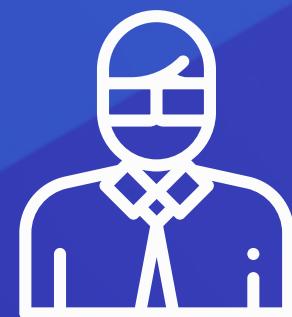




# Day 68 Sequential API

## 序列模型搭建網路



陳宇春

出題教練

# 知識地圖 深度學習簡介

## 深度學習體驗 - 啟動函數與正規化

### 深度神經網路

### Supervised Learning Deep Neural Network (DNN)

簡介 Introduction

套件介紹 Tools: Keras

組成概念 Concept

訓練技巧 Training Skill

應用案例 Application

### 卷積神經網路

### Convolutional Neural Network (CNN)

簡介 introduction

套件練習 Practice with Keras

訓練技巧 Training Skill

電腦視覺 Computer Vision

### 深度學習套件介紹

### Tools of DNN: Keras

Keras簡介與安裝

Keras 內建資料集下載

如何用 Keras 搭建類神經網路

# 本日知識點目標

- 了解 Keras Sequential API
- 了解 Keras Sequential API 與其應用的場景

# 序列模型

- 序列模型是多個網路層的線性堆疊。
- Sequential 是一系列模型的簡單線性疊加，可以在構造函數中傳入一些列的網路層：

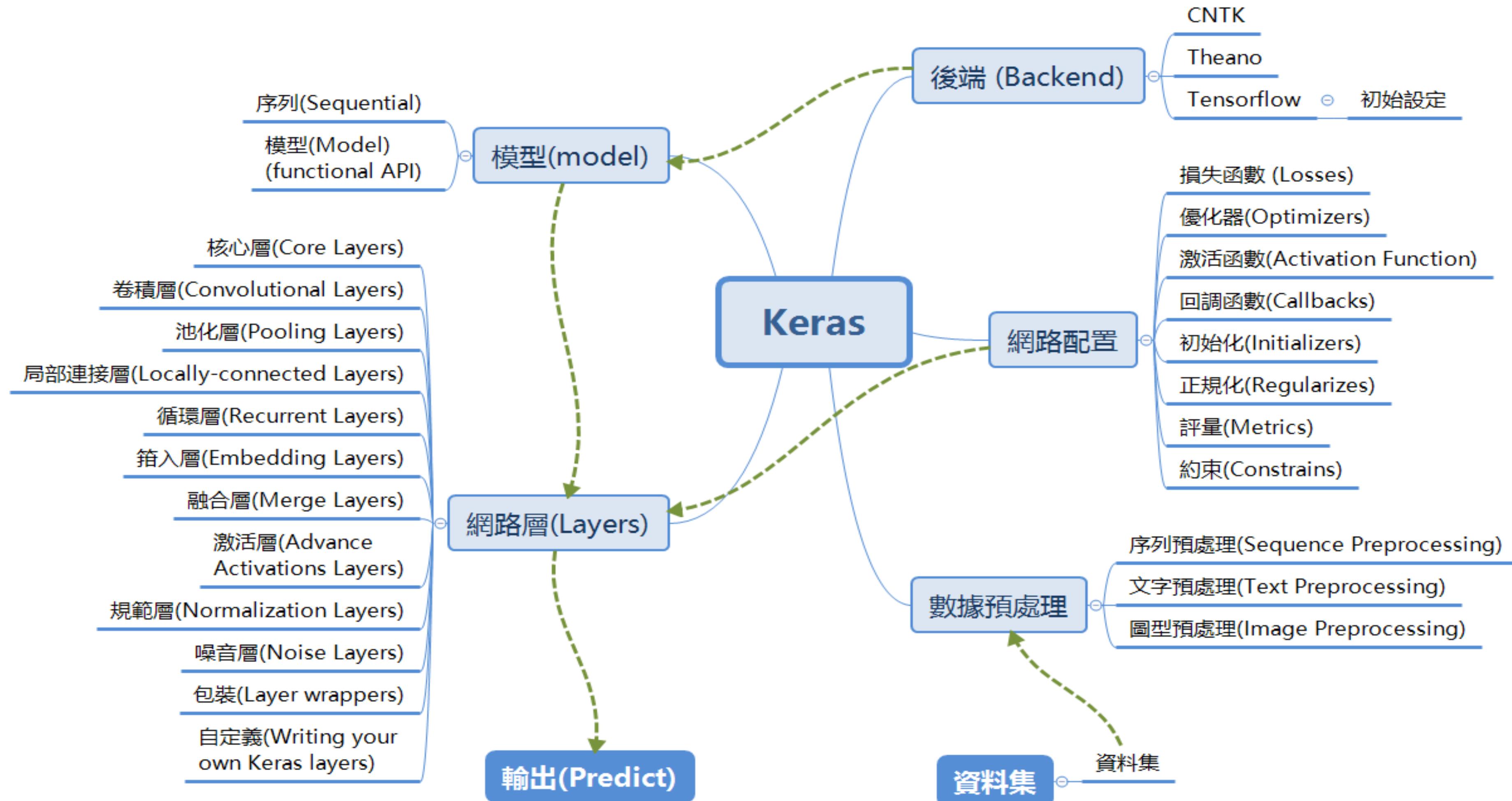
```
from keras.models import Sequential  
from keras.layers import Dense, Activation
```

```
model = Sequential([Dense(32, _input_shape=(784,)), Activation("relu")])
```

- 也可以透過 .add

```
model = Sequential()  
model.add(Dense(32, _input_dim=784))  
model.add(Activation("relu"))
```

# Keras框架回顧



# 指定模型的輸入維度

- Sequential 的第一層(只有第一層，後面的層會自動匹配)需要知道輸入的 shape
  - 在第一層加入一個 input\_shape 參數，input\_shape 應該是一個 shape 的 tuple 資料類型。
  - input\_shape 是一系列整數的 tuple，某些位置可為 None
  - input\_shape 中不用指明 batch\_size 的數目。

# 指定模型的輸入維度

- 2D 的網路層，如 Dense，允許在層的構造函數的 `input_dim` 中指定輸入的維度。
- 對於某些 3D 時間層，可以在構造函數中指定 `input_dim` 和 `input_length` 來實現。
- 對於某些 RNN，可以指定 `batch_size`。這樣後面的輸入必須是(`batch_size`, `input_shape`)的輸入

# 常用參數說明

名稱	作用	原型參數
Dense	實現全連接層	Dense(units,activation,use_bias=True, kernel_initializer='glorot_uniform', bias_initializer='zeros')
Activation	對上層輸出應用激活函數	Activation(activation)
Dropout	對上層輸出應用 dropout 以防 止過擬合	Dropout(ratio)
Flatten	對上層輸出一維化	Flatten()
Reahape	對上層輸出 reshape	Reshape(target_shape)

# 前述流程 / python 程式 對照

```
1 # build our CNN model
2 model = Sequential()
3 model.add(Conv2D(64, (3, 3), padding='same',
4                 input_shape=x_train.shape[1:]))
5 model.add(Activation('relu'))
6 model.add(Conv2D(128, (3, 3)))
7 model.add(Activation('relu'))
8 model.add(MaxPooling2D(pool_size=(2, 2)))
9 model.add(Dropout(0.25))
10
11 model.add(Flatten())
12 model.add(Dense(512))
13 model.add(Activation('relu'))
14 model.add(Dropout(0.5))
15 model.add(Dense(num_classes))
16 model.add(Activation('softmax'))
17
```

# 重要知識點複習

- Sequential 序貫模型序貫模型為最簡單的線性、從頭到尾的結構順序，一路到底
- Sequential 模型的基本元件一般需要：
  - Model 宣告
  - model.add，添加層；
  - model.compile, 模型訓練；
  - model.fit，模型訓練參數設置 + 訓練；
  - 模型評估
  - 模型預測



# 延伸 閱讀

- Getting started with the Keras Sequential model(英文)

# 推薦延伸閱讀

## Getting started with the Keras Sequential model

### Github連結

- Keras文檔的來源位於此目錄中。我們的文檔使用由MkDocs實現的擴展Markdown。
- 構建文檔 安裝MkDocs： `pip install mkdocs`
- `pip install -e .` 確保Python將導入您修改後的Keras版本。
- 從根目錄cd進入docs/文件夾並運行：
  - `KERAS_BACKEND=tensorflow python autogen.py`
  - `mkdocs serve` # 啟動本地Web服務器： localhost : 8000
  - `mkdocs build` # 在site/目錄中構建靜態站點
- Keras中文文檔 連結

### Keras Documentation

The source for Keras documentation is in this directory. Our documentation uses extended Markdown, as implemented by [MkDocs](#).

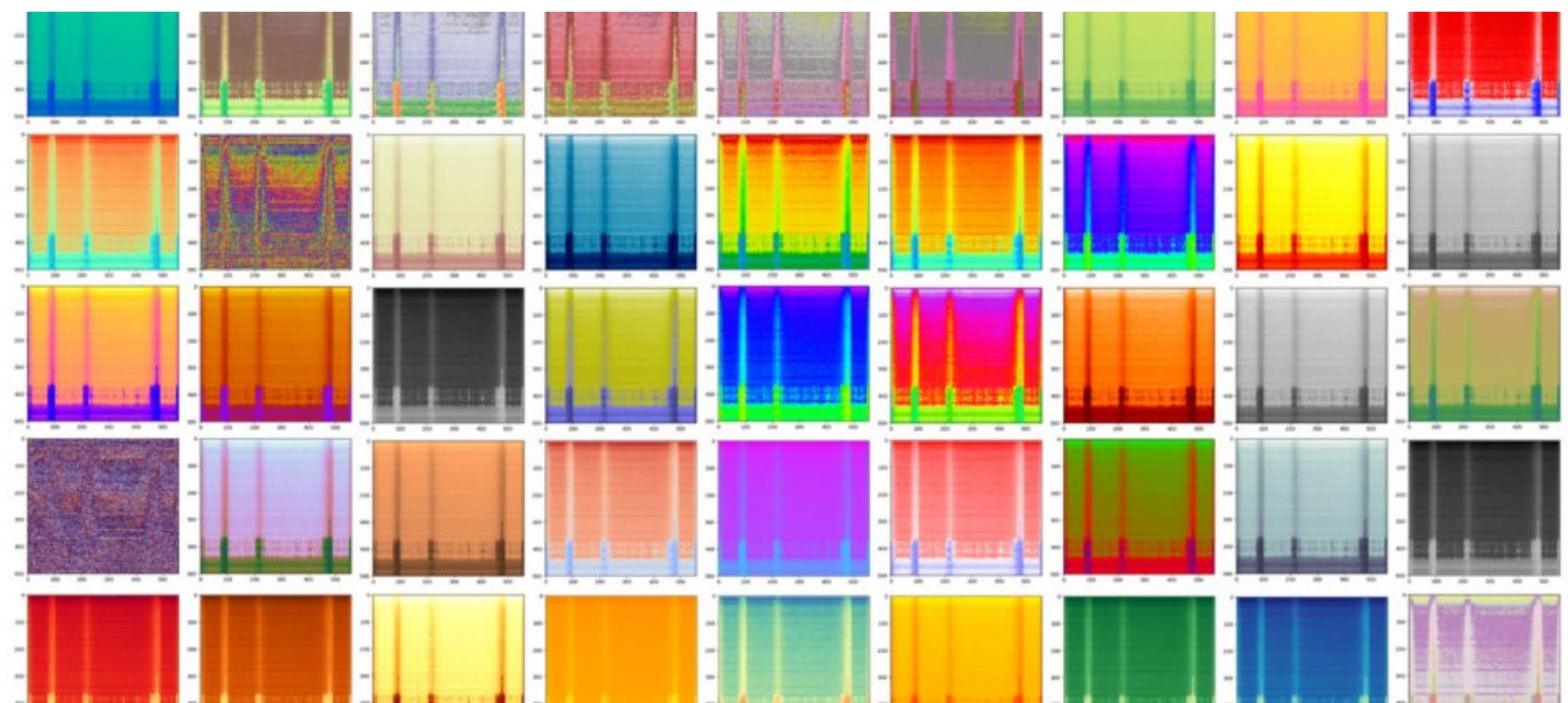
#### Building the documentation

- Install MkDocs: `pip install mkdocs`
- `pip install -e .` to make sure that Python will import your modified version of Keras.
- From the root directory, `cd` into the `docs/` folder and run:
  - `KERAS_BACKEND=tensorflow python autogen.py`
  - `mkdocs serve` # Starts a local webserver: `localhost:8000`
  - `mkdocs build` # Builds a static site in `site/` directory

# 推薦延伸閱讀

## 模型編譯

- 在訓練模型之前，您需要配置學習過程，的英文這通過 `compile`方法完成的它接收三個參數：
  - 優化器`optimizer`。它可以是現有優化器的字符串標識符，如`rmsprop`或`adagrad`，也可以是`Optimizer`類的實例。
  - 損失函數的損失，模型試圖最小化的目標函數它可以是現有損失函數的字符串標識符，如。`categorical_crossentropy`或`mse`，也可以是一個目標函數
  - 評估標準指標。對於任何分類問題，你都希望將其設置為`metrics = ['accuracy']`。評估標準可以是現有的標準的字符串標識符，也可以是自定義的評估標準函數。





解題時間

It's Your Turn

請跳出PDF至官網Sample Code & 作業  
開始解題

