

Sequential API

序列模型搭建網路



出題教練

陳宇春

本日知識點目標

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

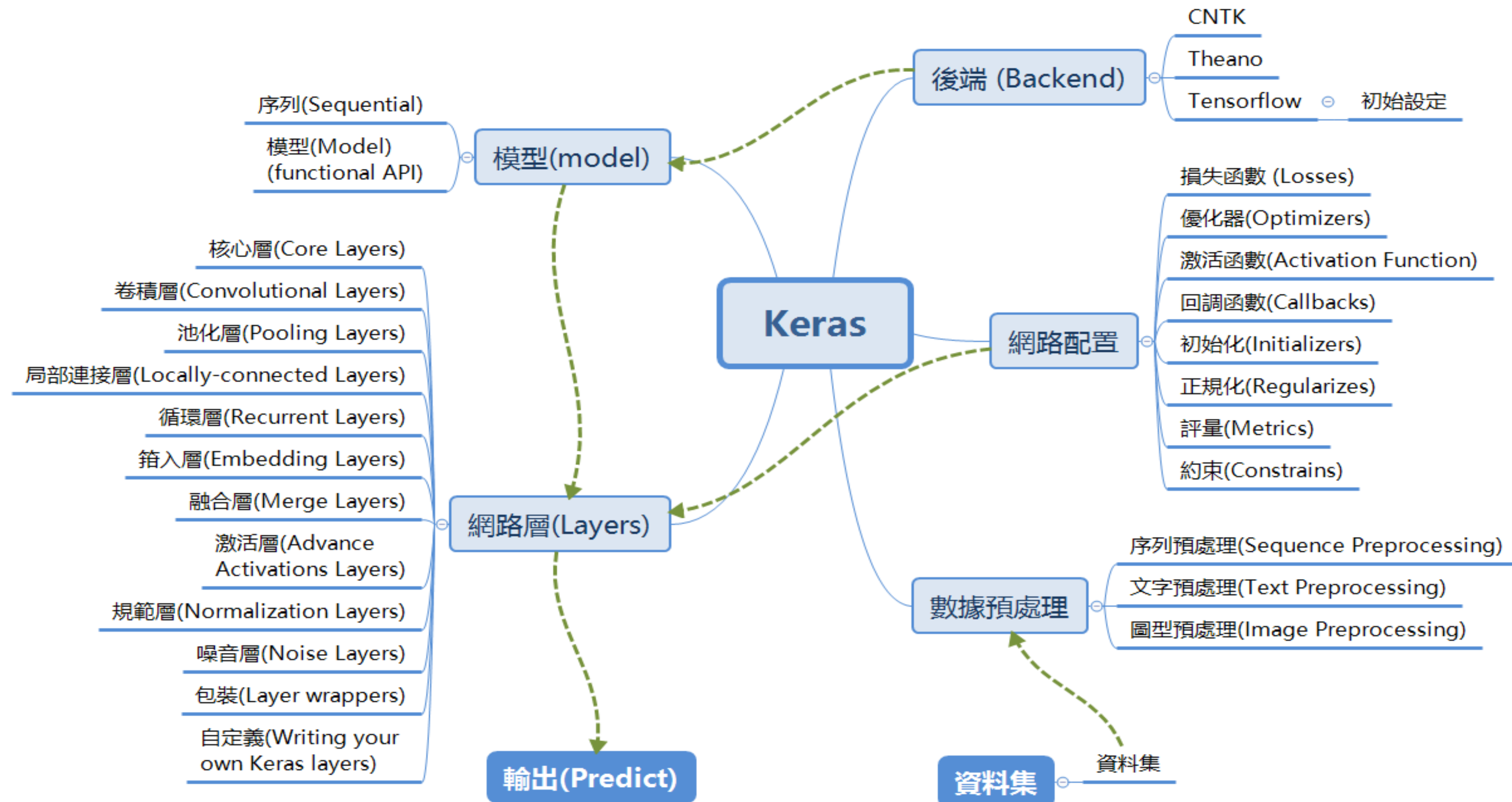
- 序列模型是多個網路層的線性堆疊。
- Sequential 是一系列模型的簡單線性疊加，可以在構造函數中傳入一些列的網路層：

```
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"))
```



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

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
```


- 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

解題時間 It's Your Turn

請跳出PDF至官網Sample Code & 作業
開始解題

