



Day 34

# 深度學習與電腦視覺 學習馬拉松

cupay 陪跑專家：陳穗碧



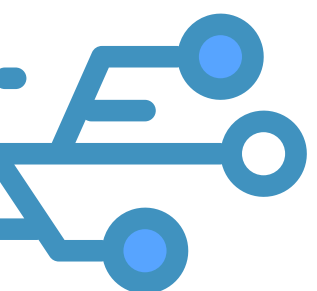


# YOLO 細節理解 – 損失函數

# 重要知識點



- 定義什麼是損失函數
- YOLOv1 損失函數的設計架構
- YOLOv1損失函數超參數的定義



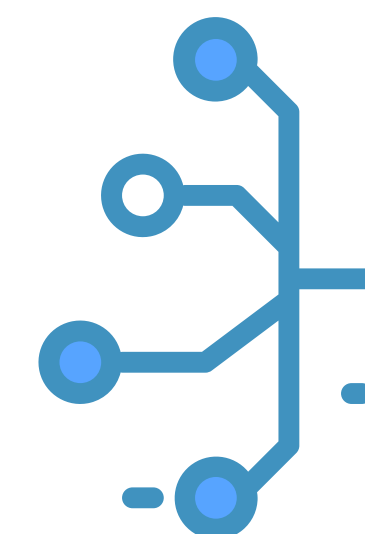
# 損失函數的定義

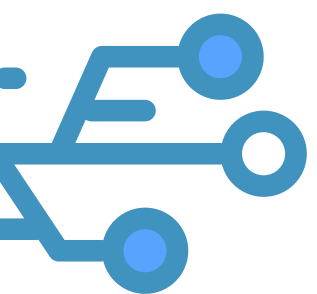


首先，我們先定義一下甚麼是損失函數，

損失函數是描述模型預測出來的結果和實際的差異的依據，期望模型預測出來結果和實際差異越小越好，別名又稱「目標函數(Object function)」，離目標越接近越好。

損失函數是一個模型的靈魂，損失函數設計的好模型會更穩定。

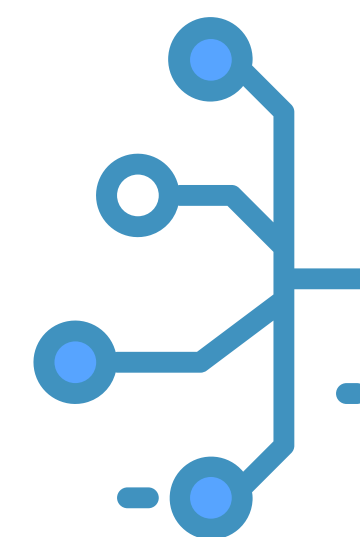


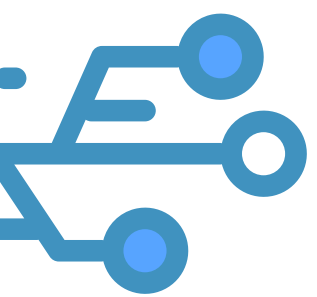


# YOLO損失函數



YOLO為one-stage的物體辨識，其損失函數的架構如下：





# YOLO損失函數

*loss function =*

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(b_{xi} - \hat{b}_{xi})^2 + (b_{yi} - \hat{b}_{yi})^2]$$

1.bbox的中心座標計算損失

$$+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{b_{wi}} - \sqrt{\hat{b}_{wi}})^2 + (\sqrt{b_{hi}} - \sqrt{\hat{b}_{hi}})^2]$$

2.bbox的寬高座標計算損失

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i)$$

3-1.bbox 有物件信心度計算損失

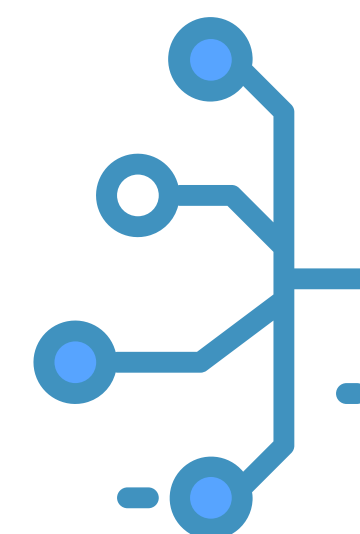
$$+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)$$

3-2.bbox 無物件信心度計算損失

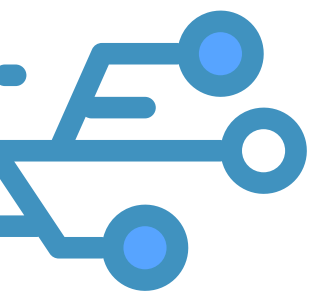
$$+ \sum_i^{S^2} 1_i^{obj} - \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))$$

4.bbox物件類別計算損失

接下來，我們會一步一步的拆解~







# YOLO 超參數意義



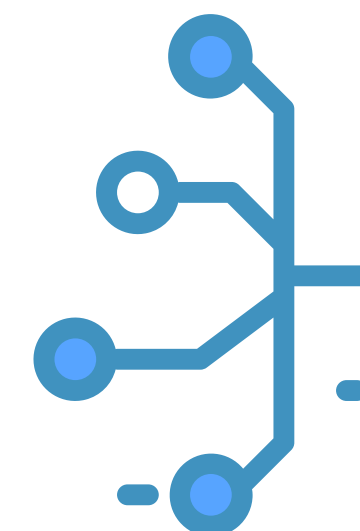
模型**超參數**是模型外部的配置，在每次執行演算法前就必須事先設定好的數值。

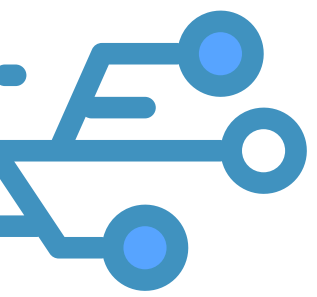
首先，我們介紹 YOLO loss中使用的超參數和參數的定義：

*loss function =*

$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(b_{xi} - \hat{b}_{xi})^2 + (b_{yi} - \hat{b}_{yi})^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{b_{wi}} - \sqrt{\hat{b}_{wi}})^2 + (\sqrt{b_{hi}} - \sqrt{\hat{b}_{hi}})^2] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i) \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i) \\ & + \sum_i^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c)) \end{aligned}$$

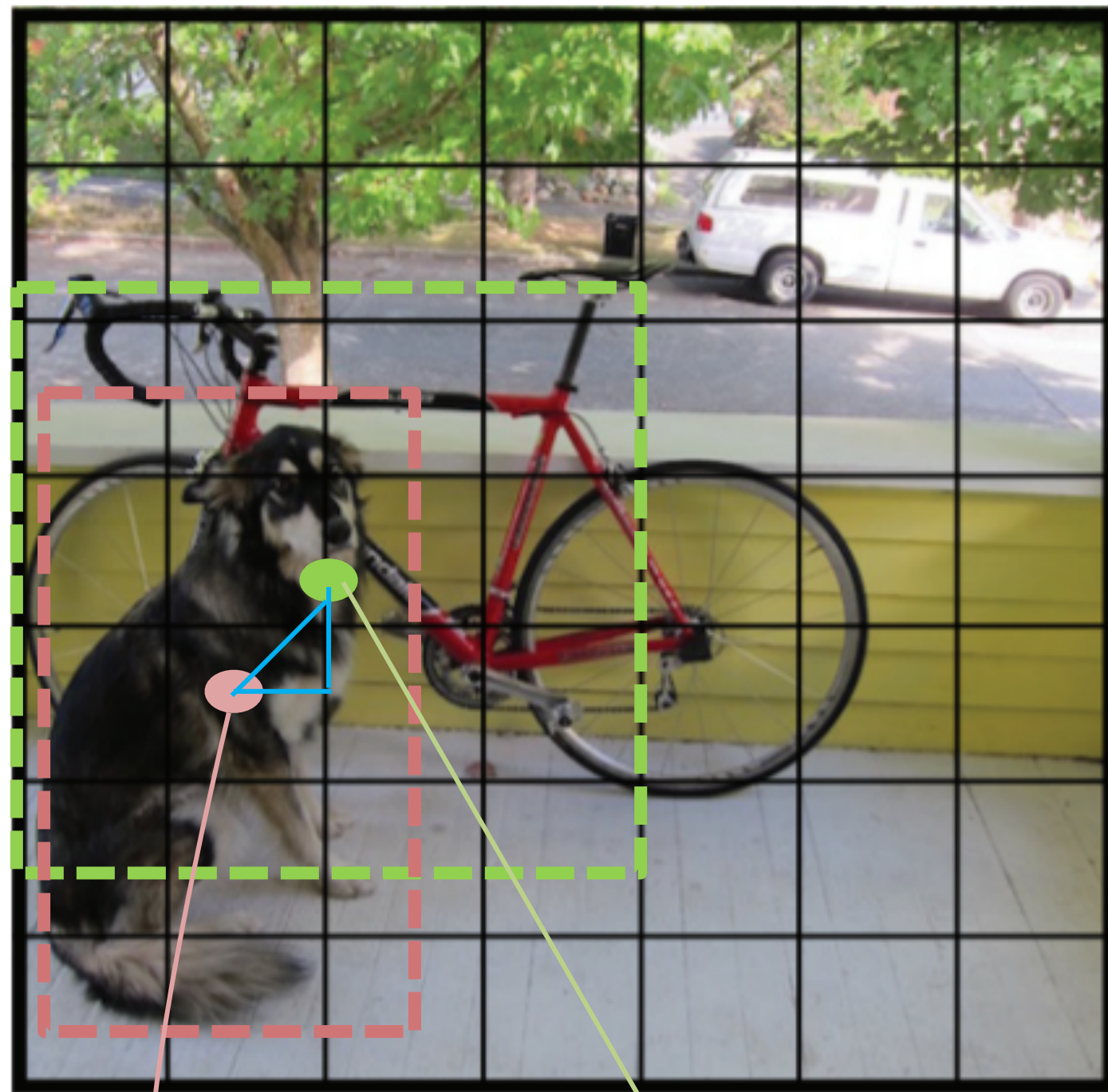
|             |                 |
|-------------|-----------------|
|             | Grid cell 總數    |
|             | 每個 cell 所含box總數 |
| Lamda_coord | (內定5)           |
| Lamda_noobj | (0.5)           |





# 1.bbox的中心座標計算損失

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(b_{xi} - \hat{b}_{xi})^2 + (b_{yi} - \hat{b}_{yi})^2]$$



$(b_{xi}, b_{yi})$

真實

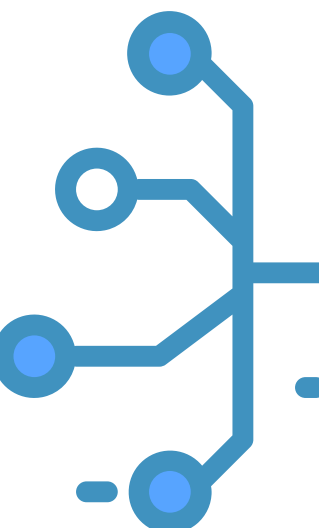
$(\hat{b}_{xi}, \hat{b}_{yi})$

預測

- $(b_{xi}, b_{yi})$  為物體標記框 ( ground truth ) 中心
- $(\hat{b}_{xi}, \hat{b}_{yi})$  為物體預測框中心
- $(b_{xi} - \hat{b}_{xi})^2 + (b_{yi} - \hat{b}_{yi})^2$  以距離定義物體標記框和物體預測框的差異量
- $1_{ij}^{obj}$ : 斷第  $i$  個網格中第  $j$  個 bbox 是否有匹配到 Groundtruth 中的某個標記框，即有該標記框與該 bbox 有最大的 IoU，則此bbox負責預測該物體，則  $1_{ij}^{obj} = 1$



$(y - \hat{y})^2$ : 真實與預測的平方和，稱為誤差平方和[sum of square error (SSE)]

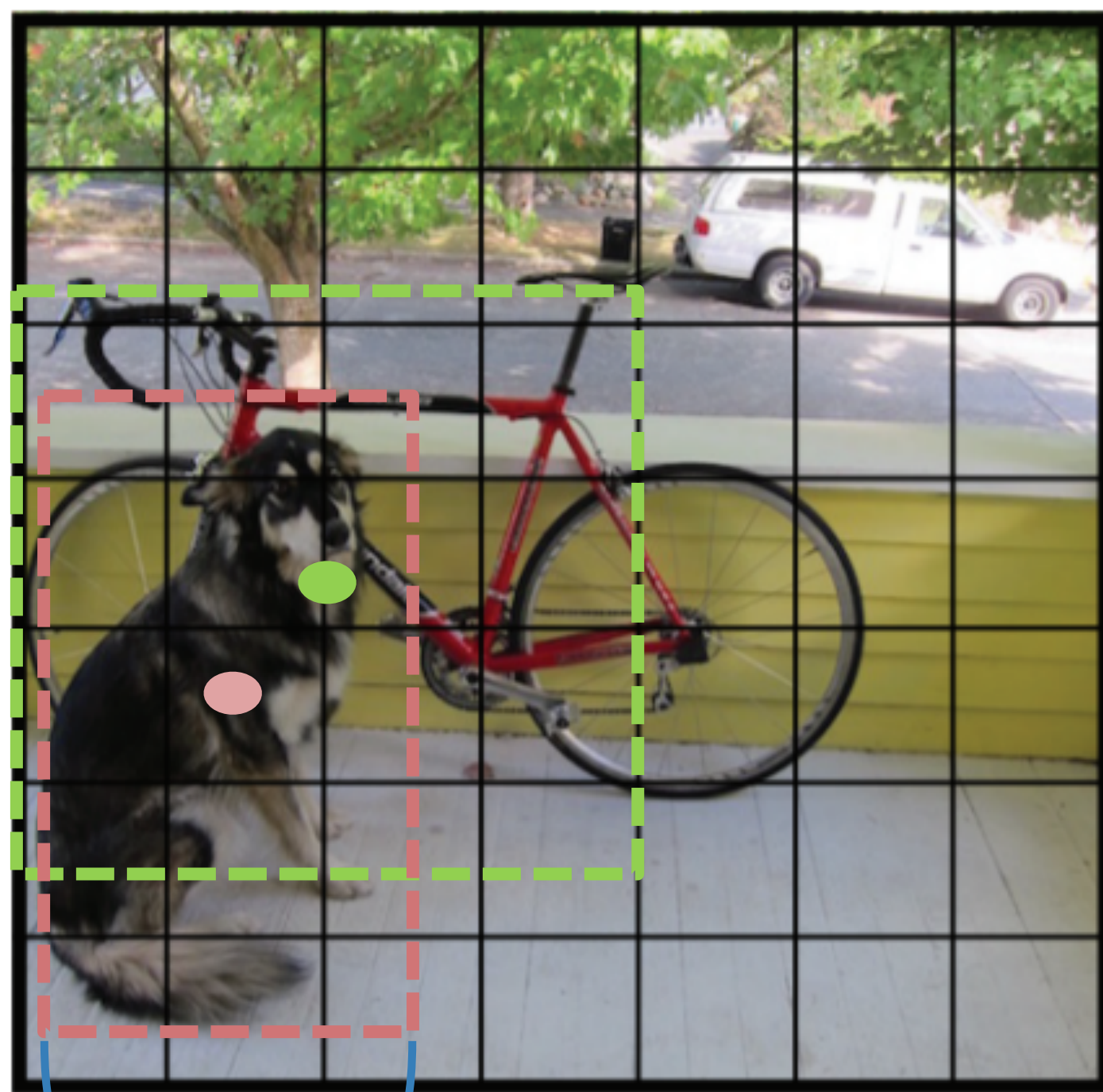






## 2. bbox的寬高計算損失

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{b_{wi}} - \sqrt{\hat{b}_{wi}})^2 + (\sqrt{b_{hi}} - \sqrt{\hat{b}_{hi}})^2]$$



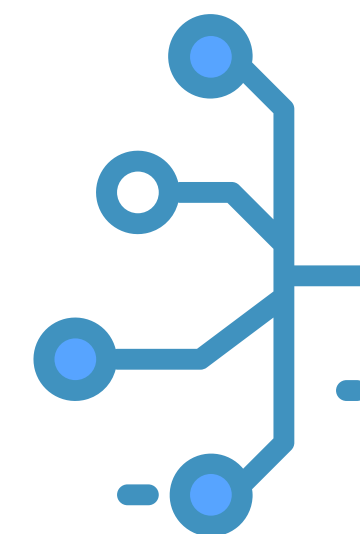
$(b_{wi}, b_{hi})$

真實

$(\hat{b}_{wi}, \hat{b}_{hi})$

預測

- $(b_{wi}, b_{hi})$  為物體標記框 ( ground truth ) 寬高
- $(\hat{b}_{wi}, \hat{b}_{hi})$  為物體預測框
- $(\sqrt{b_{wi}} - \sqrt{\hat{b}_{wi}})^2 + (\sqrt{b_{hi}} - \sqrt{\hat{b}_{hi}})^2$  描述物體標記框長寬和物體預測框長寬的差異量





### 3. bbox 物件信心度計算損失

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i) \\ + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i)$$

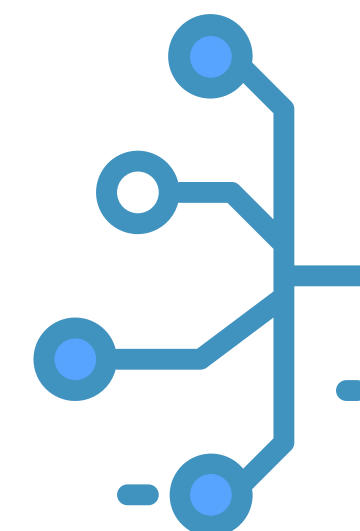
- $1_{ij}^{noobj}$ : 第 i 個 Grid 的第 j 個 bbox 中無物體為 1

有物體與無物體的 grid cell 有不同的權重，

- ① 有物體的物件信心度誤差係數為 1
- ② 無物體的物件信心度誤差係數( $\lambda_{noobj}$ )內定為 0.5



這樣的設計  
驅使模型專注於有物體的辨識能力要好，相較而言比較不在意無物體辨識成  
有物體，同時降低模型找到無物件的情形。



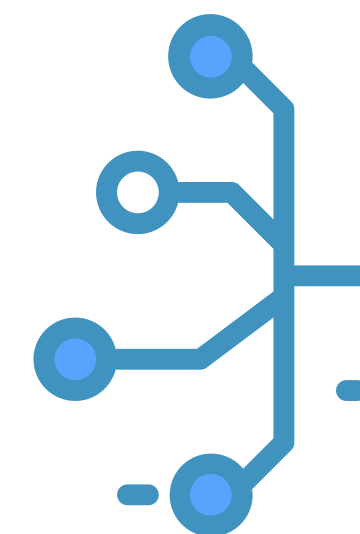


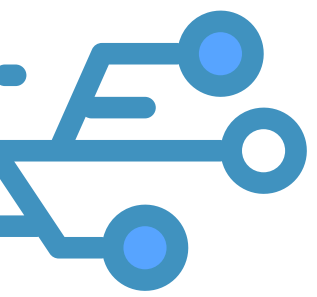


## 4. 對bbox預測的類別計算損失

$$+ \sum_i^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))$$

- $1_i^{obj}$ : 判斷是否有某一物體標記框中心落在網格i 中，
- 網格中心包含有該物體的中心，就負責預測該物體的類別機率。
- 只針對有物體的grid cell 計算類別損失





# $\lambda$ : 讓重要的值重要，無足輕重的值則折扣其影響力

*loss function =*

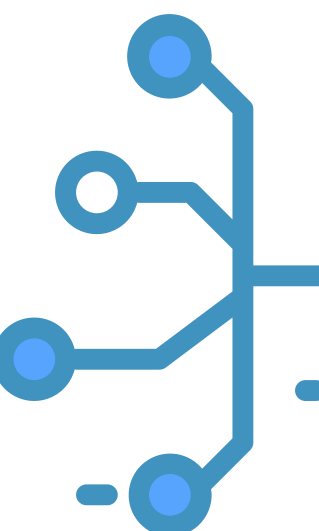
$$\begin{aligned} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(b_{xi} - \hat{b}_{xi})^2 + (b_{yi} - \hat{b}_{yi})^2] \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{b_{wi}} - \sqrt{\hat{b}_{wi}})^2 + (\sqrt{b_{hi}} - \sqrt{\hat{b}_{hi}})^2] \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} (C_i - \hat{C}_i) \\ & + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} (C_i - \hat{C}_i) \\ & + \sum_i^{S^2} 1_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c)) \end{aligned}$$

|  |       |
|--|-------|
|  | (內定5) |
|  | (0.5) |



在第7張投影片，我們知道YOLO，  
 $\lambda_{coord}$  內定為5， $\lambda_{noobj}$  內定為 0.5，  
這樣的設定讓模型擁有樣的能力？

不同超參數代表模型有不同的辨識能力，內定的超參數代表模型著重於物體定位的能力，相較比較不在乎無物體的grid cell辨識成有物體的情形。





# 知識點 回顧

- 損失函數是描述模型預測出來的結果和實際的差異的依據
- YOLO 損失函數的設計包含物件位置的定位與物件類別辨識
- YOLO損失函數透過超參數設定模型有不同的辨識能力



- [超參數的詳細說明](#)
- [不同型態的損失函數介紹](#)

## 機器/深度學習: 基礎介紹-損失函數 (loss function)



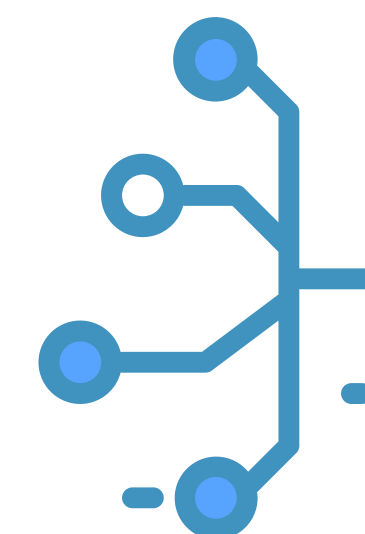
Tommy Huang [Follow](#)

Sep 27, 2018 · 13 min read



機器學習大部分的算法都有希望最大化/最小化一個函數/指標，這個函數被稱為「目標函數(Object function)」(如果修過最佳化理論對這個名詞應該不陌生，整本書都在講這個)，比如K-means群集分析就是希望「最小化群內的資料和群心的誤差平方和」，PCA則是希望「投影向量讓投影後的資料變異量最大化」。不同的演算法用到的假設都不太一樣，所以目標函數都不一樣。最近很夯的人工智慧(幾乎都是深度學習)用到的目標函數基本上都是「損失函數(loss function)」，而模型的好壞有絕大部分的因素來至損失函數的設計。

損失函數基本上可以分成兩個面向(分類和回歸)，基本上都是希望最小化損失函數。





# 解題時間 Let's Crack It



請跳出 PDF 至官網 Sample Code & 作業開始解題