深度學習 - 第五章

Status	Book
Created time	@December 25, 2024 1:44 PM

第五章

Google Colab 檔案

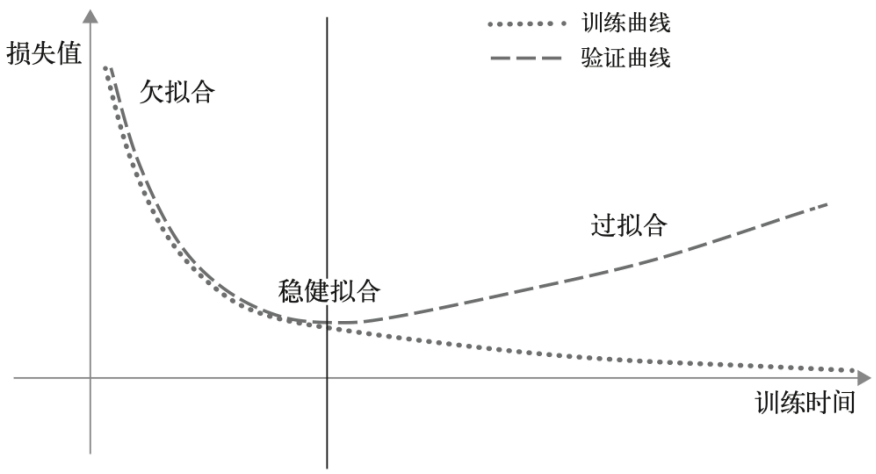
Ch5.ipynb

5-1 普適化：機器學習的終極目標

- 優化是利用訓練資料來不斷強化模型。
- 普適化是指訓練完成的模型，在未見過資料上的表現。
- 過度配適也稱過度擬合，學到訓練資料獨有特徵，但未普遍出現在其他資料，導致普適化能力降低。

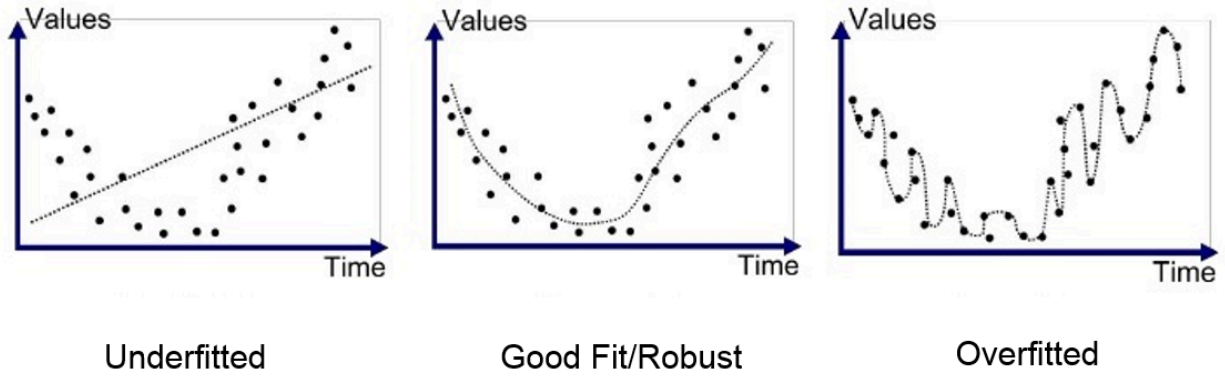
5-1-1 低度配適與過度配適

- 訓練初期驗證表現會不斷提升。
- 訓練一段時間後驗證表現無可避免地走下坡。

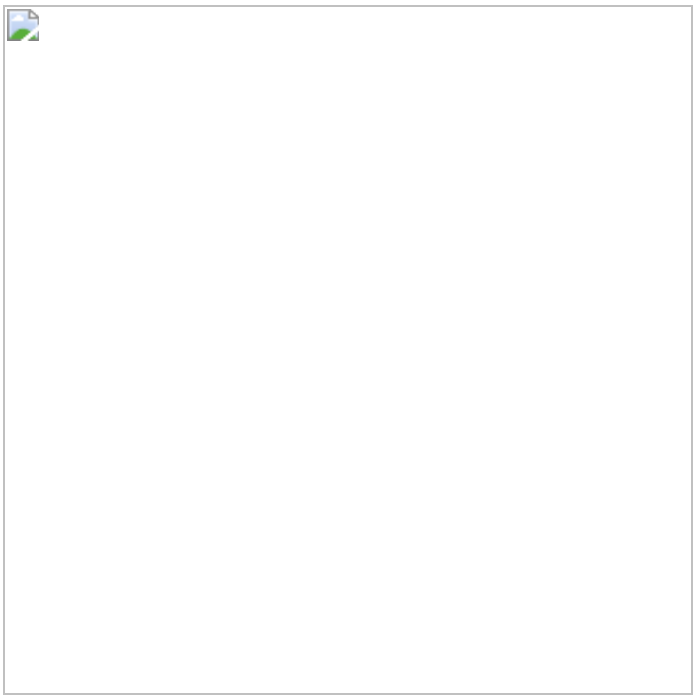


- 過度配適最常發生在具有雜訊（noisy）的資料，資料中具有不確定性，或是出現次數很少的特徵。

- 更糟糕的還有圖片的標籤是錯誤的。
- 訓練過程中，模型針對這些離群值（outlier）學習，普適化自然會下降。



- 模糊特徵
  - 當問題本身具備不確定性或模稜兩可，就算是字跡清晰、標籤正確的資料也有可能是雜訊。
  - 許多答案具有隨機性，同樣的數據未必有相同的結果，中間仍有變動的可能性。
  - 模型對特徵空間中模糊地帶的資料學習太過深入，容易出現過度配適的問題。
  - 比較穩健（robust）的模型會忽略訓練資料中個別的資料點，並從大處著眼。
- 罕見特徵（rare feature）與虛假關聯（spurious correlation）
  - 使用罕見特徵的資料集來訓練，很可能出現過度配適。
  - 並非只有罕見特徵會出現虛假關聯。
  - 若資料集有 54% 正面、46% 負面語意，模型可能將差異的 8% 視為普遍存在的差異。



- 兩組資料蘊含相同的有效特徵，訓練出來的模型在驗證準確度卻有差距，差距來自於虛假關聯。
- 當你加入越多雜訊（亂數），準確度就會越低。
- 訓練前進行特徵挑選（feature selection），常見是對特徵計算分數，保留分數在閥值之上的特徵。

5-1-2 普適化在深度學習中的本質

- 把 MNIST 資料集的標籤打亂，重新訓練一次。

```
Epoch 100/100
375/375 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - accuracy: 0.9017 - loss: 0.3301 - val_accuracy: 0.0974 - val_loss: 8.0408
```

- 模型參數夠多，就可以成功擬合隨機資料，最終模型會直接把輸入和標籤之間的關係死背起來。
- 流形假說（manifold hypothesis）
    - 所有自然資料在其所處的高維度空間中，都可以被排列（編碼）到一個低維度的流形上。
    - 流行假說表示

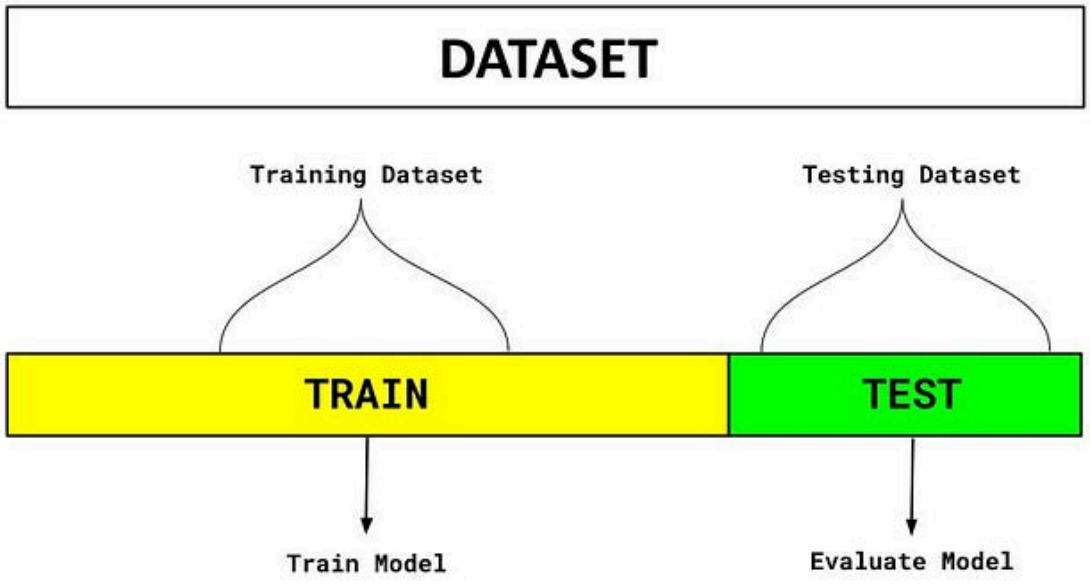
        - 機器學習模型只需擬合（學習）輸入空間中的潛在流形（latent manifold）即可，潛在子空間中的資料相對比較簡單、低維度、且具有高結構體。
        - 在這些流形中，兩樣本間必然可進行內插（interpolate），可以沿著一條連續的路徑將一個樣本漸變成另一個樣本，該路徑上所有樣本都位在流形內。
    - 樣本間內插特性就是普適化能力的關鍵，訓練樣本涵蓋範圍夠廣即可，不用很密。
- 以內插法作為普適化的基礎
    - 靠空間中有限樣本來理解空間的整體性（totality），用內插法填滿其中空白。
    - 潛在流形的內插法與原始空間中的線性內插法不一樣
        - 流形內插法：潛在流行空間上的中間點。
        - 線性內插法：直接在編碼空間上進行算術平均。
    - 潛在流形內插法是由原始空間萃取出來的低維度子空間，資料具備集中且連續漸變的特性。
    - 內插法只能理解那些與已見過事物非常相近的對象，實現局部普適化（local generalization）。
    - 對那些與已見過事物不太相近的對象，其實也有可能普適化。
    - 認知機制（cognitive mechanism）
        - 抽象化、符號化、邏輯推論、建立常識、先驗知識等，通常統稱理性。
        - 有別於本質比較接近內插法的直覺、樣式認知（pattern recognition）。
- 為何深度學習能運作
    - 深度學習本質上就是刻畫出一個巨大、複雜的曲線（流形），並逐步調整參數，直到擬合大部分的資料點。
    - 要擬合的資料並不是遍佈整個空間的稀疏獨立點，而是處於輸入空間內部的高度結構化、低維度流形，這就是流形假說的內容。
    - 模型曲線用梯度下降法來擬合訓練資料的流形時，曲線總是漸進且平滑地變動，因此在訓練中存在中間點（intermediate point），該點大致逼近普遍資料的自然流形（達到穩健擬合狀態）。
    - 深度學習除了擁有足夠表徵能力，也非常適合用來學習潛在流形

        - 輸入與輸出間建立一個連續且平滑的映射關係。
        - 設計良好的模型可以結構化地映射訓練資料的資訊形狀（the “shape” of information）。
- 訓練資料是一座必須跨越的大山
    - 普適化能力主要取決於資料的自然結構，而非模型的特性。
    - 提升普適化表現，資料篩選（data curation）和特徵工程（feature engineering）不可或缺。
    - 要模型表現良好，最好可以在輸入空間中密集抽樣（dense sampling）。
        - 訓練資料應密集涵蓋整個輸入空間的流形，尤其在決策邊（decision boundary）附近。
    - 無法取得更多資料時，調降模型所能容納的資訊，或在擬合曲線上加入限制，讓模型只能記憶很有限或很常見得態樣，優化過程中就會強迫模型專注在最突出的態樣，這樣比較有可能提升普適化能力 → 常規化（regularization）。

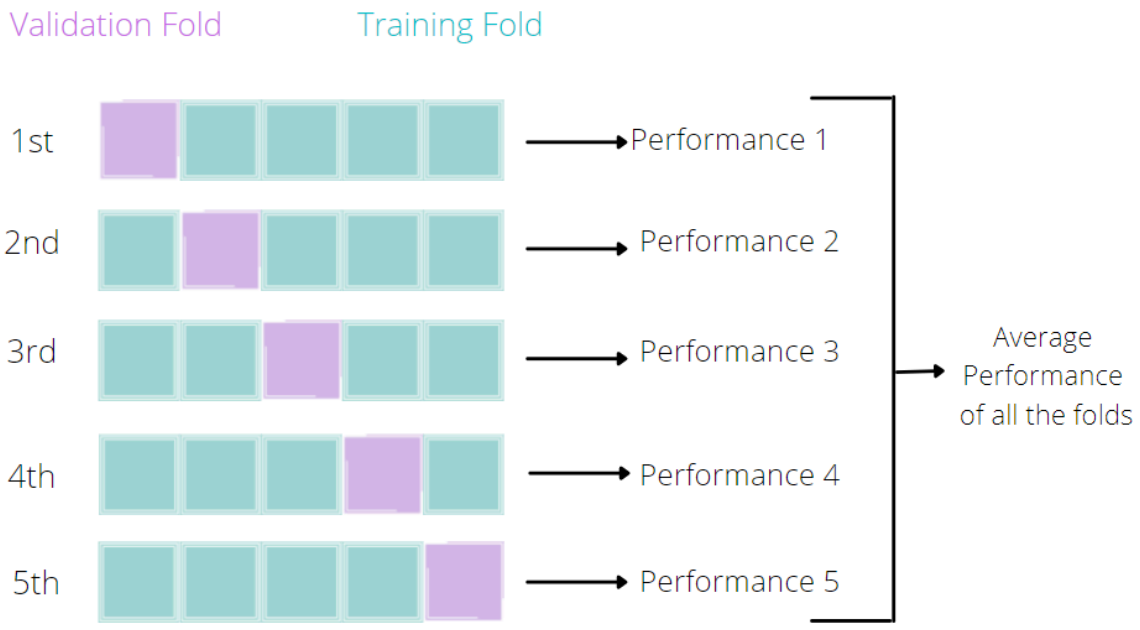
5-2 評估機器學習模型

5-2-1 訓練集、驗證集和測試集

- 用訓練集來訓練模型，驗證集來評估模型，測試集進行最後的評估測試。
- 每個模型要有多少層（深度），或每一層的規模要多大（多少個神經單元，即寬度）等等，這些稱為模型的超參數（hyperparameter），跟神經網路的權重參數（weight parameter）不同。
- 將模型在驗證資料上的表現，作為回饋資訊來調整模型的超參數。
    - 根據驗證集表現，在超參數空間中尋找最佳配置。
    - 可能對驗證集過度配適（overfitting to the validation set）→ 資料洩漏（information leak）。
- 簡單拆分驗證（Simple holdout validation）
    - 訓練集切出一部份資料作為驗證集，用來調整模型超參數。
    - 可用資料很少，驗證集合測試集的樣本也會很少，導致統計代表性不足。
        - 重新洗牌後，訓練出的模型表現差異很大，那多半表示手上的資料太少了。



- K 折驗證（K-fold validation）
  - 拆分成大小相同的 K 個區塊 → 輪流選取區塊作為驗證集 → 其餘區塊來重新訓練模型 → 保存該次訓練驗證分數。
  - 經過 K 次訓練後，取分數平均值為最終分數，參照此調整模型超參數。
  - 模型表現會因資料隨機拆分產生顯著差異時，K 折驗證法可適時解決這個問題。



- 多次洗牌的 K 折驗證（Iterated K-fold validation with shuffling）
  - 適用於資料量不足，且需要盡可能精確地驗證（評估）模型的情況。
  - 多次應用 K 折驗證，每次分割區塊前重新洗牌，最終驗證分數是所有驗證分數的平均值。
  - 每次要訓練和評估 n * K 個模型，運算成本高很多。

5-2-2 打敗基準線

- 基準線（baseline）
  - 模型訓練唯一可以得到的回饋就是驗證分數。
  - 選擇大致的基準線，以此為超越的目標，模型表現超過基準線，就可以繼續往下走。
  - 基準線可以是任何分類器的準確度，或其他任意非機器學習技巧的表現分數。
  - 基本表現達不到，要麼用了錯誤的模型，要麼是問題不適合直接用機器學習來解決。

5-2-3 模型評估時的注意事項

- 資料代表性（data representativeness）
  - 訓練資料和測試集都有一定代表性，足以反映資料的分佈。
  - 拆分為訓練集和測試集前，通常需要對資料隨機洗牌（randomly shuffle），使兩者有一定代表性。
- 時間的方向性（the arrow of time）
  - 試圖從過去資料預測未來狀態，就不應該打亂資料，這樣會造成時間漏失（temporal leak）。
  - 確保測試資料的發生時間是在訓練資料之後。
- 資料中的重複現象（redundancy in your data）
  - 若某些資料點出現兩次，打亂並拆分後可能導致訓練集和驗證集中出現相同資料，使用相同資料進行訓練與驗證，導致模型表現不可信。
  - 須確保訓練集和驗證集之間沒有交集。
- 若想找可靠的方法評估模型表現，首先該思考

    - 如何監看優化與普適化。
    - 低度配適與過度配適之間的張力變化。

5-3 提升模型的擬合表現

- 訓練出能展現基本普適化能力，且會發生過度配適的模型，再開始專注解決過度配適問題。
- 這個階段會遇到 3 個問題
    - 訓練沒有成效，損失值始終降不下來。
    - 訓練成效尚可，沒展現出普適化能力，甚至連基準線都無法超越。
    - 訓練損失與驗證損失都隨時間降低，表現也比基準線好，但無法達到過度配適，模型仍處於低度配適階段。

5-3-1 調整梯度下降的關鍵參數

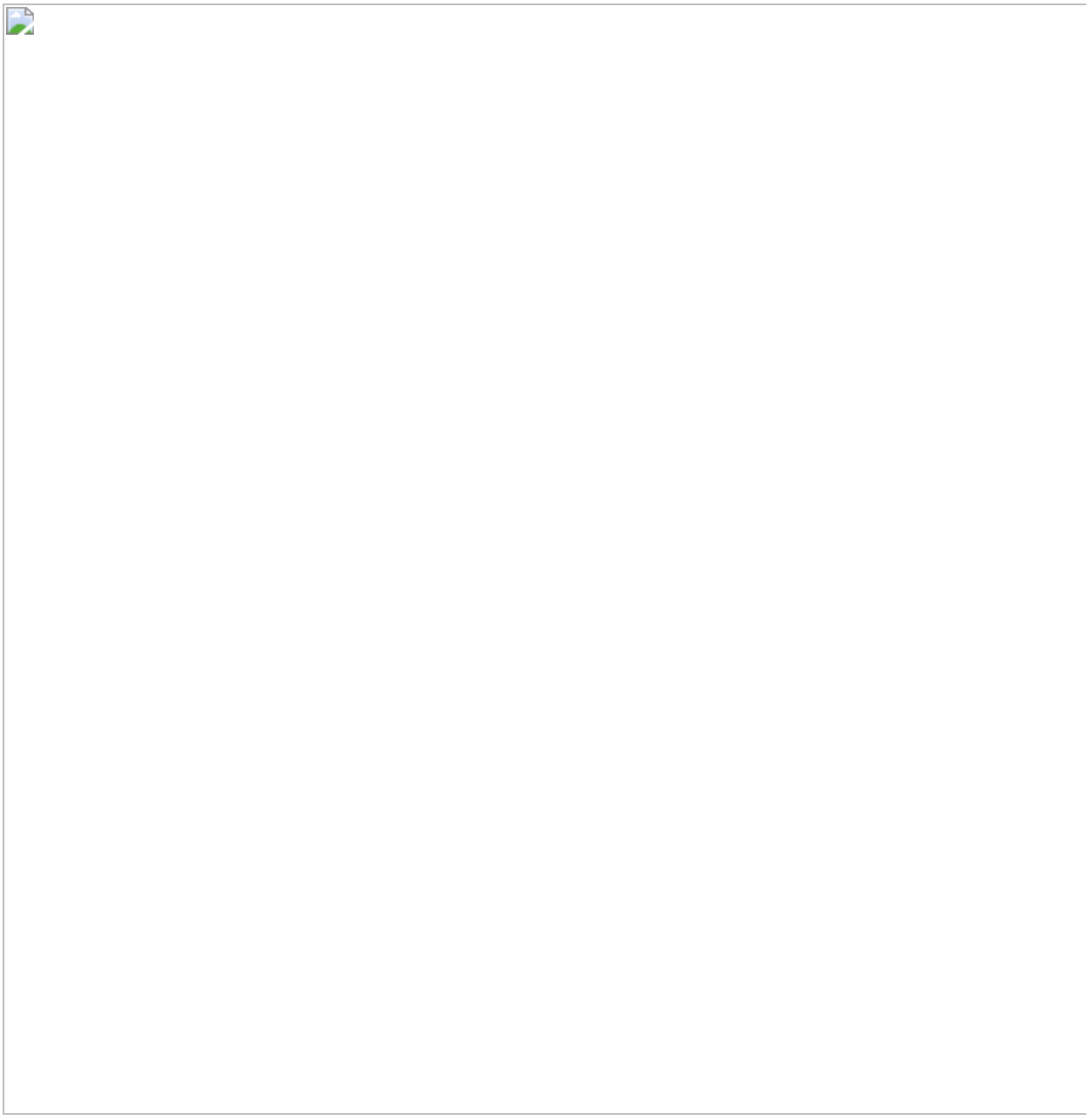
- 損失值無法下降時 → 梯度下降參數配置出了問題。
    - 優化器、權重的初始分佈、學習率或者批次量。
    - 參數彼此相關，通常調整學習率或批次量就足夠了，其他參數當作常數。
- 降低或提高學習率
    - 學習率太高，一不小心就超過最佳值，參數更新幅度太大，損失值擺盪，無法收斂到最低點。
    - 學習率太低，訓練進展太慢，誤以為訓練卡住。
- 增加批次量
    - 批次中有更多的樣本，提供更多資訊，雜訊較少（較低的變異量）。

5-3-2 利用既有的架構

- 模型擬合訓練資料，但驗證準確度無法提升，模型有在學習，但始終無法普適化。
    - 訓練樣本中資訊不足以預測目標值，目前設定的問題是無解的。
    - 模型不適合處理該問題
        - 為特定問題選擇合適的模型架構，對實現普適化來說是必要的

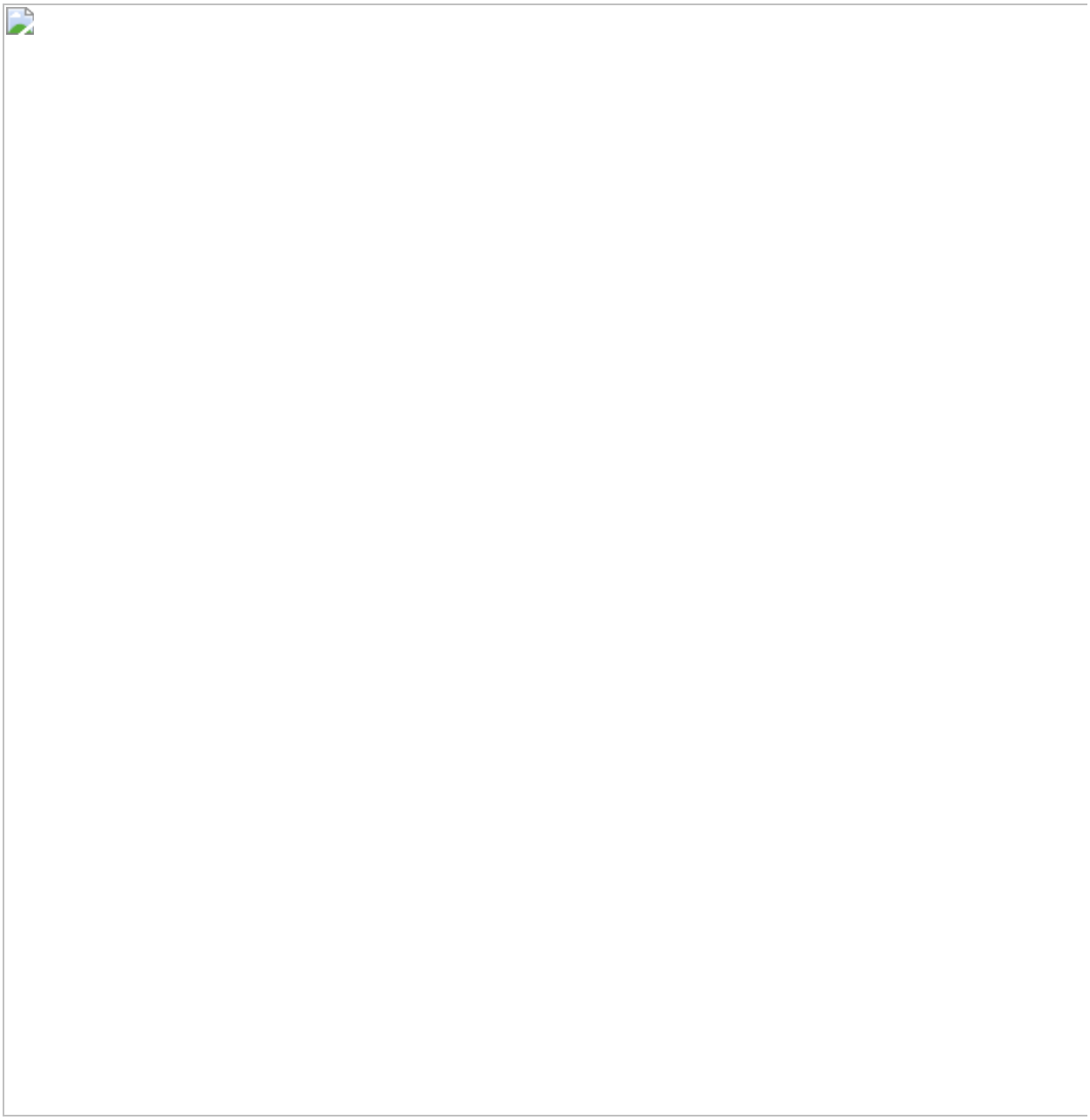
5-3-3 提升模型容量（capacity）

- 模型擬合資料，驗證損失在下降，模型已具有某個程度的普適化能力，開始過度配適。



- 驗證損失已經停滯不前，並無反轉跡象，始終無法達到過度配適的程度。
- 一定有辦法讓模型過度配適，不會過度配適，可能就是表徵能力（representational power）不足。

- 也許需要更大的模型來容納更多資訊。
- 提升表徵能力
    - 增加神經層數量。
    - 更大的神經層（有較多神經單元）。
    - 選擇更適合當下問題的神經層類型（選用更好的模型架構）。



- 一開始快速下降，約 8 個週期後開始上升，代表發生過度配適。

5-4 提高普適化能力

- 模型具備一定普適化能力，且開始過度配適時，就可以專注在提升普適化能力了。

5-4-1 資料集篩選（Dataset curation）

- 確保有足夠資料，更多的資料通常可以得到更好的模型。
- 減少標註上的錯誤，將資料視覺化來觀察是否出現異常值（anomalies），並仔細檢查標籤。
- 清理資料並處理缺失值。
- 有很多特徵，不知道哪些是有用的，先做特徵選擇。

5-4-2 特徵工程（feature engineering）

- 透過更簡單的方式來表示問題，使問題更容易處理。
- 讓潛在流形更平滑、更簡單、更有組織性。
- 深度學習減少了大多數特徵工程的需求，但仍需要特徵工程
    - 良好的特徵可以在使用更少資源的狀況下，更有效的解決問題。
    - 良好的特徵能用更少的資料解決問題。

5-4-3 使用早期停止（early stopping）

- 在 Keras 中的經典做法，使用 EarlyStopping 回呼（callback），一旦 callback 程式發現驗證指標不再繼續提升，就停止訓練，把最佳的模型狀態儲存下來。

5-4-4 將模型常規化

- 常規化可以避免模型過度擬合訓練資料，讓模型在驗證階段表現更好，具有較佳的普適化能力。
- 縮減神經網路規模
    - 模型記憶資源有限時，將很難保存太多訓練樣本與目標值之間的對應關係，模型採用萃取過的資料表示法，以建立對目標的預測能力。
    - 模型應該擁有足夠的參數來避免低度配適，不該過度缺乏記憶資源。

- 在容量過大（too much capacity）和容量不足（not enough capacity）之間取得平衡。
- 通常從較少的層數和參數開始，逐漸增加層的大小或增加新的層，直到驗證損失不再進步為止。

- 較小模型過度配適時間比原始模型來得晚，過度配適後，表現變差的程度比較慢。

    - 模型在訓練一開始就發生過度配適，驗證損失曲線波動很大，就代表模型太大了（也可能是驗證過程不可靠，驗證集太小）。
    - 模型容量越大，對訓練資料的學習速度就越快，但過度配適的可能性也越大。
- 加入權重常規化（weight regularization）
    - 採用較少的權重值以限制模型的複雜性，讓權重值分佈更為常規化（regularized），透過對損失函數中較大的權重加上代價（cost）項來實現，模式通常有兩種
        - L1 常規化（L1 regularization）：代價項和權重的絕對值（權重的 L1 norm）成正比。
        - L2 常規化（L2 regularization）：代價項和權重的平方（權重的 L2 norm）成正比。
            - 也稱權重衰減（weight decay），數學上的權重衰減是等同於 L2 常規化的。
        - Keras 做常規化，只有在訓練時使用，驗證時自動拿掉，損失函數中的 Wi 就更小。

L1 regularization on least squares:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} |w_i|$$

L2 regularization on least squares:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^{k} w_i^2$$

    - 在 Keras 中，只要指名參數把權重常規化物件傳入神經網路層就可以了。

  - l2（0.002）表示該層權重矩陣中，每個權重值都會加上「(0.002*權重值）的平方」到模型的總損失值上。
  - 懲罰（penalty，即代價項），只會在訓練階段加入，訓練階段損失值會比其他階段高。
  - L2 常規化的模型變得比原始模型更能抵抗過度配適。
  - 權重常規化一般用在較小的深度學習模型上。
- 加入丟棄法（dropout）

- 在訓練期間隨機丟棄神經網路層的一些輸出特徵（把特徵值設為 0）。
- 丟棄率（dropout rate）只要被歸零的特徵比例，通常介於 0.2 到 0.5 之間。
- 測試階段並不會丟棄任何特徵，取而代之的是層的輸出值將依照丟棄率的比例縮小，以平和訓練時特定輸出被歸零的影響。

- 加入 Dropout 層的模型與原始模型，效果有明顯改善。
- 與加入 L2 常規化模型相比，表現似乎也更好。