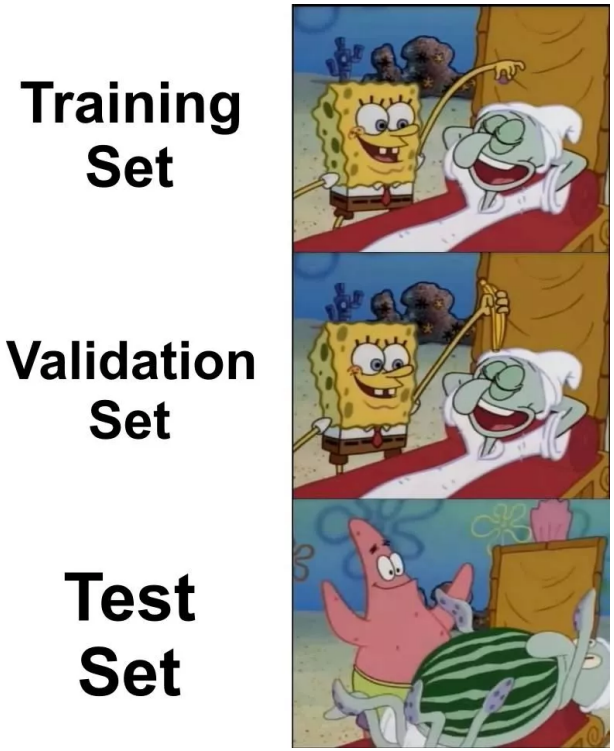
防止模型過擬合的必備方法!

Mahitha 智能算法 昨天

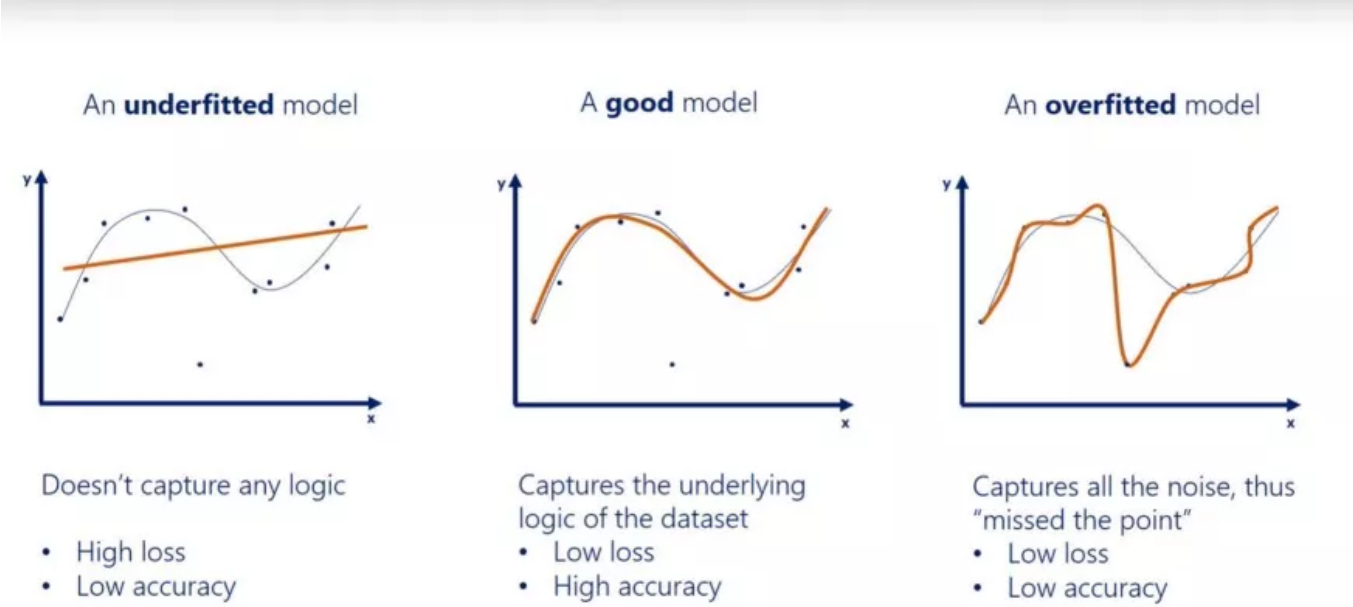
來自｜機器之心 作者｜Mahitha
鏈接｜ https://mahithas.medium.com/overfitting-identify-and-resolve-df3e3fdd2860

> 正如巴菲特所言：「近似的正確好過精確的錯誤。」

在機器學習中，過擬合（overfitting）會使模型的預測性能變差，通常發生在模型過於復雜的情況下，如參數過多等。本文對過擬合及其解決方法進行了歸納闡述。



在機器學習中，如果模型過於專注於特定的訓練數據而錯過了要點，那麼該模型就被認為是過擬合。該模型提供的答案和正確答案相距甚遠，即準確率降低。這類模型將無關數據中的噪聲視為信號，對準確率造成負面影響。即使模型經過很好地訓練使損失很小，也無濟於事，它在新數據上的性能仍然很差。欠擬合是指模型未捕獲數據的邏輯。因此，欠擬合模型具備較低的準確率和較高的損失。



如何確定模型是否過擬合?

構建模型時，數據會被分為3 類：訓練集、驗證集和測試集。訓練數據用來訓練模型；驗證集用於在每一步測試構建的模型；測試集用於最後評估模型。通常數據以80:10:10 或70:20:10 的比率分配。

在构建模型的过程中，在每个 epoch 中使用验证数据测试当前已构建的模型，得到模型的损失和准确率，以及每个 epoch 的验证损失和验证准确率。模型构建完成后，使用测试数据对模型进行测试并得到准确率。如果准确率和验证准确率存在较大的差异，则说明该模型是过拟合的。

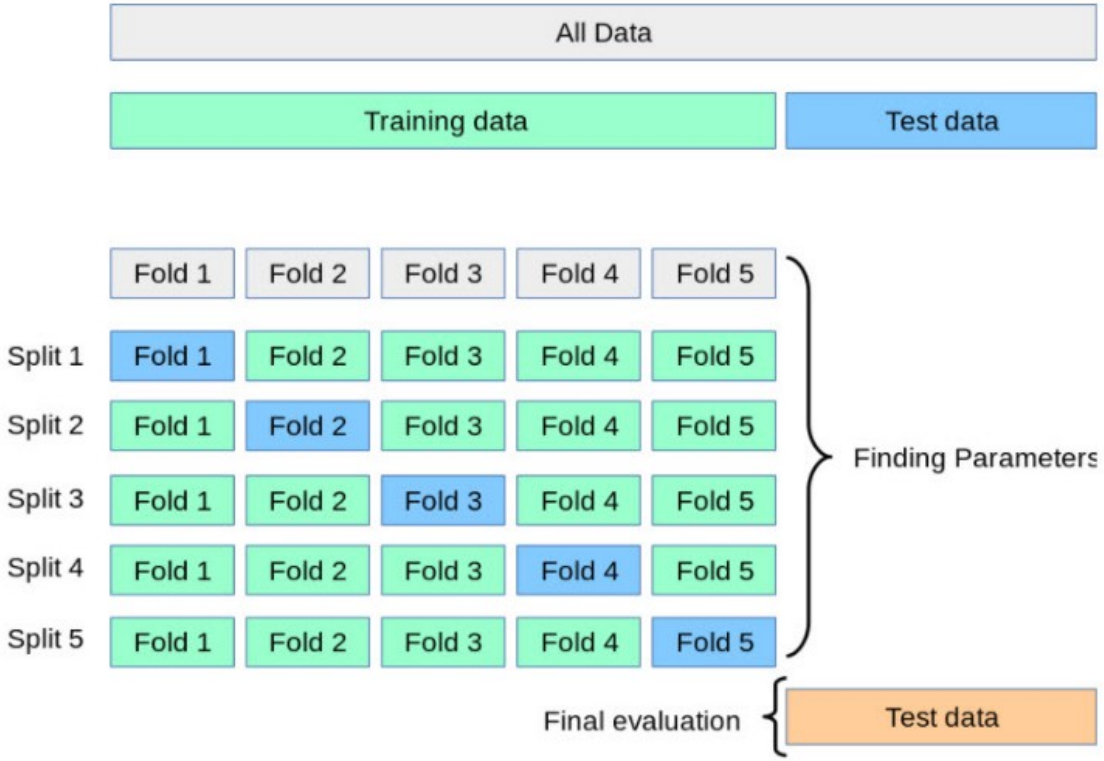
如果验证集和测试集的损失都很高，那么就说明该模型是欠拟合的。

如何防止过拟合

交叉验证

交叉验证是防止过拟合的好方法。在交叉验证中，我们生成多个训练测试划分（splits）并调整模型。K-折验证是一种标准的交叉验证方法，即将数据分成 k 个子集，用其中一个子集进行验证，其他子集用于训练算法。

交叉验证允许调整超参数，性能是所有值的平均值。该方法计算成本较高，但不会浪费太多数据。交叉验证过程参见下图：



用更多数据进行训练

用更多相关数据训练模型有助于更好地识别信号，避免将噪声作为信号。数据增强是增加训练数据的一种方式，可以通过翻转（flipping）、平移（translation）、旋转（rotation）、缩放（scaling）、更改亮度（changing brightness）等方法来实现。

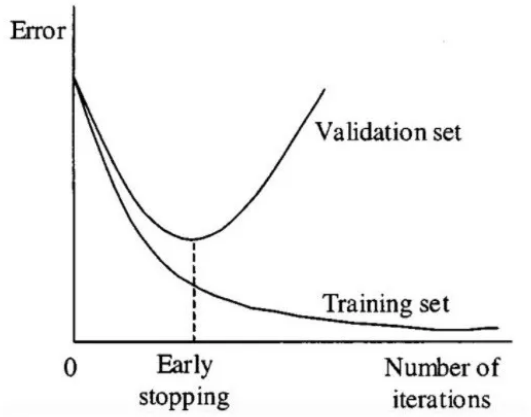
移除特征

移除特征能够降低模型的复杂性，并且在一定程度上避免噪声，使模型更高效。为了降低复杂度，我们可以移除层或减少神经元数量，使网络变小。

早停

对模型进行迭代训练时，我们可以度量每次迭代的性能。当验证损失开始增加时，我们应该停止训练模型，这样就能阻止过拟合。

下图展示了停止训练模型的时机：



正则化

正则化可用于降低模型的复杂性。这是通过惩罚损失函数完成的，可通过 L1 和 L2 两种方式完成，数学方程式如下：

$$L(x,y) \equiv \sum_{i=1}^{n}(y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^{n}|\theta_i|$$

L1 惩罚的目的是优化权重绝对值的总和。它生成一个简单且可解释的模型，且对于异常值是鲁棒的。

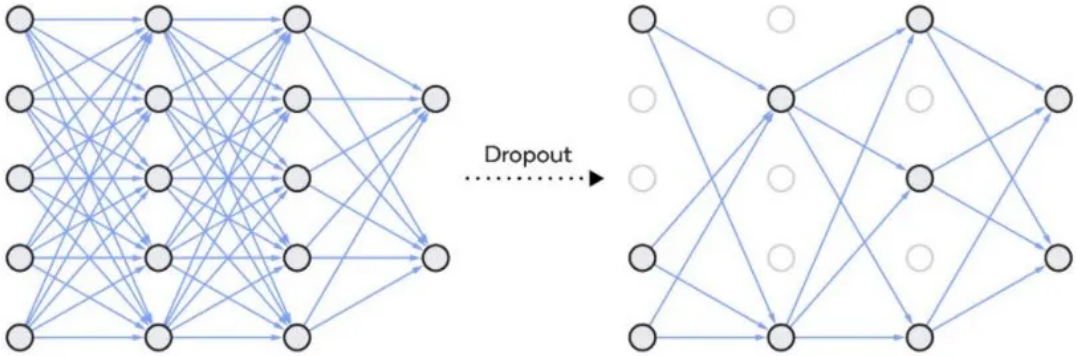
$$L(x,y) \equiv \sum_{i=1}^{n}(y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^{n}\theta_i^2$$

L2 惩罚权重值的平方和。该模型能够学习复杂的数据模式，但对于异常值不具备鲁棒性。

这两种正则化方法都有助于解决过拟合问题，读者可以根据需要选择使用。

Dropout

Dropout 是一種正則化方法，用於隨機禁用神經網絡單元。它可以在任何隱藏層或輸入層上實現，但不能在輸出層上實現。該方法可以免除對其他神經元的依賴，進而使網絡學習獨立的相關性。該方法能夠降低網絡的密度，如下圖所示：



總結

過擬合是一個需要解決的問題，因為它會讓我們無法有效地使用現有數據。有時我們也可以在構建模型之前，預估到會出現過擬合的情況。通過查看數據、收集數據的方式、採樣方式，錯誤的假設，錯誤表徵能夠發現過擬合的預兆。為避免這種情況，請在建模之前先檢查數據。但有時在預處理過程中無法檢測到過擬合，而是在構建模型後才能檢測出來。我們可以使用上述方法解決過擬合問題。

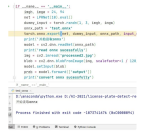
原文鏈接：https://mahithas.medium.com/overfitting-identify-and-resolve-df3e3fdd2860

—完—

喜歡此內容的人還喜歡

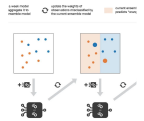
模型部署翻車記：pytorch轉onnx踩坑實錄

極市平台



常用的模型集成方法介紹：bagging、boosting 、stacking

小白學視覺



【AI+機器人】不同行走速度下的步態模型學習

TG課題組