Ch06 工作流程

2022年10月26日

上午 11:19

1.定義任務

了解問題領域和客戶要求的業務邏輯。收集數據集，了解數據代表甚麼，然後選擇衡量任務成功的方式

2.開發模型

準備你的數據以便他可以被機器學習模型處理，訓練一個具有泛化且可以過擬合的模型，然後正則化調和你的模型，直到獲得最佳的泛化性能。

3.部屬模型

開始使用模型，監控模型在新數據下的表現，並開始收集數據建構下一代模型。

6.1定義任務

如果不深入了解你正在做的事情，就無法做好工作。

為甚麼你的客戶試圖解決這個特定問題?他們將從解決方案中獲得甚麼價值?你的模型將如何使用?甚麼樣的數據可用或可以收集?

6.1.1框定問題

1.你的輸入數據是甚麼?你想預測甚麼?只有當你有可用的訓練數據十，你才能學會預測某是，例如:只有當你同時擁有電影評論和情感註釋時，你才能學會對電影評論的情感進行分類。因此**數據可用性**通常是此階段的限制因素。在許多情況下，你將不得不自己收集和註釋新數據集。

2.你面臨甚麼類型的機器學習任務?**二分類?多類分類?**

標量回歸?向量回歸?多累、多標籤分類?圖像分割?

其他的東西，比如生成或強化學習?**在某些情況下，機器學習甚至不是理解數據的最佳方式，你應該使用其他方式，例如簡單的老式統計分析。**

舉例來說

(1).照片搜尋引擎是一個多類、多標籤的分類任務。

(2).垃圾郵件檢測是一個二分類任務，任務，如果你將"攻擊性內容"設置為一個單獨的類，那就會變成三向分類任務。

(3).事實證明，音樂推薦引擎通過矩陣分解(協同過濾)處理的更好。

(4).信用卡詐欺檢測是一個二元分類任務。

(5).點擊率預測是標量回歸任務。

6.1.2收集數據集

依但你了解了任務的性質並知道你的輸入和目標是甚麼，就該進行數據收集了，這是大多數機器學習中最艱鉅、最耗時和最昂貴的部分。

你應該自己註釋數據嗎?

你應該使用像亞馬遜土耳其機器人 (Amazon Mechanical Turk) (是亞馬遜公司旗下的一個眾包網站，由亞馬遜雲計算服務提供服務，於2005年11月2日推出。有需求的企業可以在該網站上發佈遠程工作任務，例如識別出圖像或視頻中的特定內容、編寫產品描述以及回答問題等。 )這樣的眾包平台來收集標籤嗎?

你應該使用專業數據標籤公司的服務嗎?

在選擇時，需要考慮

1.數據標註者是否需要成為專家，或者任何人都可以對數據進行註釋

ExL任何人都可以對貓狗圖像分類進行標籤，但對狗品種分類的標籤需要專業知識。

2.如果註釋數據需要專業知識，你能訓練人們去做嗎?如果沒有，你如何獲得相關專家的幫助?

3.你了解專家提出註釋的方式嗎?

**謹防非代表性數據**

機器學習模型只能理解與他們之前看到的類似的輸入。因此，用於訓練的數據應能代表生產數據。

假如你在製作一個辨識菜餚名稱的模型，但來自憤怒用戶的回饋開始到來，你的應用在10次中有8次得到錯誤答案。

因為你的訓練數據中，使用的是專業品質、光線充足、令人垂涎的圖片，與實際上不同。你的訓練數據不能代表生產數據，歡迎來到機器學習地獄。

**概念飄移**

機器學習只能用於記憶訓練數據中存在的模式。

這會使模型精度逐漸下降。

例如:2013年的流行音樂與2020年的流行音樂不同

**抽樣偏差**

隨機選擇的樣本並不能代表所有數據

EX:1948美國總統大選

6.1.3了解你的數據

在開始訓練數據之前，你應該探索和可視化你的數據，深入了解釋甚麼使其具有預測性，這將為特徵工程和篩選潛在問題提供信息。

1.如果你的數據包含圖像或自然語言文本，請直接查看一些樣本(及其標籤)。

2.如果你的數據包含數字特徵，最好繪製特徵值的直方圖，以了解所取值的範圍和不同值的頻率。

3.如果你的數據包含位置信息，請將其繪製在地圖上。是否出現任何清晰的模式?

4.某些樣本是否缺失某些特徵的值?如果是這樣，你需要在準備數據時處理此問題

5.如果你的任務是分類問題，請打印數據中每個類的實例數。這些類的代表性是否大致相同?如果沒有， 您將需要考慮這種不平衡。

6.檢查目標洩漏：數據中是否存在提供有關目標的信息 且在生產中可能不可用的特徵。

6.1.4選擇衡量成功的標準

要控制某事，您需要能夠觀察它。要在項目上取得成功，您必須首先定義成功的含義。準確性？精度和召回率？客戶保留率？您的成功指標將指導您在整個項目中做出的所有技術選擇。它應該直接與您的更高級別目標保持一致，例如客戶的業務成功。

對於平衡分類問題，每個類別的可能性相同，準確度和接受者**操作特徵(ROC) 曲線下的面積（縮寫為 ROC AUC）**是常見的指標。對於類

不平衡問題、排名問題或多標籤分類，您可以使用精度和召回率，以及精度或 ROC AUC 的加權形式。

衡量成功的指標。要了解機器學習成功指標的多樣性以及它們與不同問題領域的關係，瀏覽 Kaggle 上的數據科學競賽會很有幫助（https:// kaggle.com）；它們展示了廣泛的問題和評估指標。

6.2.1開發模型

一旦你知道你將如何衡量你的進步，你就可以開始模型開發了。大多數教程和研究項目都假設這是唯一的步驟 跳過假設已經完成的問題定義和數據集收集，並跳過假設由其他人處理的模型部署和維護。事實上，模型開發只是機器學習工作流程中的一步，如果你問我，這並不是最困難的一步。機器學習中最難的事情是構建問題以及收集、註釋和清理數據。

**6.2.1準備數據**

正如您之前所了解的，深度學習模型通常不會攝取原始數據。數據預處理旨在使手頭的原始數據更適合神經網絡。這包括**矢量化、標準化或處理缺失值**。

許多預處理技術是特定於域的（例如，特定於文本數據或

圖像數據）；當我們在實際示例中遇到它們時，我們將在接下來的章節中介紹它們。現在，我們將回顧所有數據域共有的基礎知識。

**矢量化**  
神經網絡中的所有輸入和目標通常必須是浮點數據的張量（或者，在特定情況下，整數或字符串的張量）。**無論您需要處理什麼數據，**

**聲音、圖像、文本 你必須首先變成張量，這一步稱為數據向量化。**例如，在第4 章的前兩個文本分類示例中，**我們從表示為整數列表（代表單詞序列）的文本開始，然後使用 one‑hot 編碼將它們轉換為float32數據的張量**。在數字分類和預測房價的例子中，數據是矢量化的，所以我們可以跳過這一步。

**標準化**

在第 2 章的 MNIST 數字分類示例中，我們從編碼為 0‑255 範圍內的整數的圖像數據開始，對灰度值進行編碼。在將這些數據輸入網絡之前，我們必須**將其轉換為float32並除以 255，這樣我們最終會得到 0‑1 範圍內的浮點值**。

同樣，在預測房價時，我們從具有各種範圍的特徵開始 一些特徵具有較小的浮點值，而另一些具有相當大的整數值。在我們將這些數據輸入我們的網絡之前，我們獨立地對每個特徵進行歸一化，使其標準差為 1，平均值為 0。

一般來說，向神經網絡輸入具有相對較大值的數據（例如，多位整數，其遠大於網絡權重所採用的初始值）或異構數據是不安全的（例如，一個特徵在 0‑1 範圍內而另一個特徵在 100‑200 範圍內的數據）。這樣做會觸發大的梯度更新，從而阻止網絡收斂。為了使您的網絡更容易學習，您的數據應具有以下特徵：

1.取小值 ‑ 通常，大多數值應在 0–1 範圍內。

2.同質化 所有特徵的取值都應在大致相同的範圍內。

此外，以下更嚴格的規範化實踐很常見並且可以提供幫助，儘管它並不總是必要的

獨立地對每個特徵進行歸一化，使其均值為 0。

獨立地對每個特徵進行歸一化，使其標準差為 1。

使用numpy可以輕易達成

x -= x.mean(axis=0) #1

x /= x.std(axis=0)

您的數據中有時可能會缺少值。例如，在房價示例中，第一個

特徵（數據中索引 0 的列）是人均犯罪率。

如果此功能不適用於所有樣本怎麼辦？然後，您將在訓練或測試數據中缺少值。

您可以完全放棄該功能，但不一定必須這樣做。

如果該特徵是分類的，則可以安全地創建一個表示“缺少值”的新類別。該模型將自動學習這對目標意味著什麼。

如果特徵是數字的，請避免在特徵形成的潛在空間中輸入任意值，例如“0”，這會使訓練的模型更難泛化。相反，請考慮將缺失值替換為特徵的平均值或中值。

在數據集中。您還可以訓練模型以根據其他特徵的值來預測特徵值。

請注意，如果您期望測試數據中缺少分類特徵，但網絡是在沒有任何缺失值的數據上訓練的，那麼網絡將不會學會忽略缺

失值！在這種情況下，您應該人為地生成缺少條目的訓練樣本： 多次復制一些訓練樣本，並丟棄一些您認為可能在測試數據中丟失的分類特徵。

6.2.2選擇評估指標

正如您在前一章中所了解的，模型的目的是實現泛化，並且您在整個模型開發過程中做出的每一個建模決策都將受到旨在衡量泛化性能的驗證指標的指導。您的驗證協議的目標是準確估計您的成功指標（如準確性）將取決於實際生產數據。

該過程的可靠性對於構建有用的模型至關重要。

在第 5 章中，我們回顧了三種常見的評估協議：

1.維護一個保留驗證集 當你有大量數據時，這是要走的路。

2.進行 K 折交叉驗證 當您的樣本太少而無法進行

可靠驗證時，這是正確的選擇。

3.進行迭代 K 折驗證 這是為了在可用數據很少時執行高度準確的模型評估。

6.2.3擊敗基線

當您開始研究模型本身時，您的初始目標是實現統計功效，正如您在第 5 章中所看到的：即開發一個能夠超越簡單基線的小型模型。

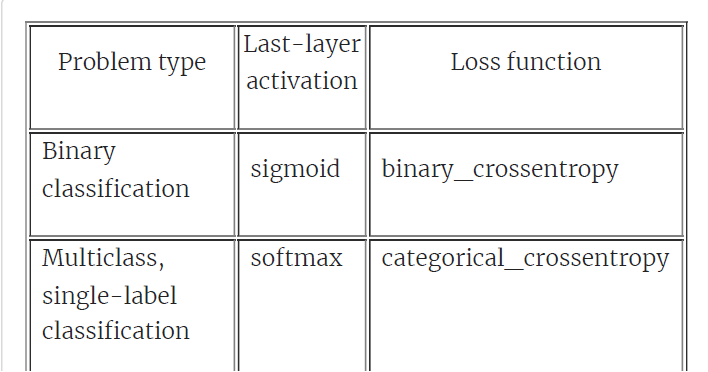
在這個階段，這些是您應該關注的最重要的三件事：

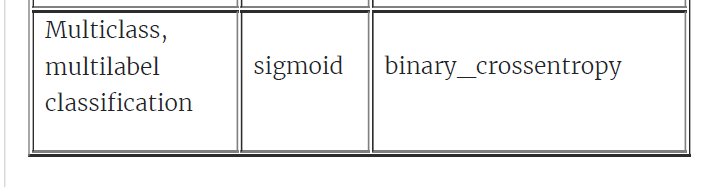
特徵工程 過濾掉無信息的特徵（特徵選擇）並利用您對問題的了解來開發可能有用的新特徵。

選擇正確的架構先驗 您將使用哪種類型的模型架構？一個密集連接的網絡，一個卷積網絡，一個循環神經網絡，一個變壓器？深度學習甚至是完成任務的好方法，還是應該使用其他方法？選擇一個足夠好的訓練配置 你應該使用什麼損失函數？什麼批量大小和學習率？

通常不可能直接針對衡量問題成功與否的指標進行優化。**有時沒有簡單的方法將度量轉化為損失函數**；畢竟，損失函數需要在只給定小批量數據的情況下是可計算的（理想情況下，損**失函數應該可以計算到一個數據點）並且必須是可微分的（否則，您不能使用反向傳播來計算訓練你的網絡**）。例如，**廣泛使用的分類度量 ROC AUC 不能直接優化。**

因此，**在分類任務中，通常會針對 ROC AUC 的代理指標進行優化，例如交叉熵。在一般來說，您可以希望交叉熵越低，ROC AUC 越高。**





6.2.5正則化和調整你的模型

一旦你的模型能夠過擬合，你就知道你走在正確的道路上，此時，你的目標變成最大化泛化性能。

此階段將花費最多時間：您將反復修改模型、訓練它、評估您的驗證數據（此時不是測試數據）、再次修改它並重複，直到模型盡可能好得到。以下是您應該嘗試的一些事情：

1.嘗試不同的架構；添加或刪除圖層。添加輟學。

2.如果您的模型很小，請添加 L1 或 L2 正則化。

嘗試不同的超參數（例如每層的單元數或優化器的學習率）以找到最佳配置。

3.可選地，迭代數據管理或特徵工程：收集和註釋

更多數據，開發更好的特徵，或刪除似乎沒有信息的特徵。

過使用自動超參數調整軟件（例如 KerasTuner ），可以自

動化大部分工作。我們將在第 13 章中介紹這一點。

請注意以下幾點：每次使用來自驗證過程的反饋來調整模型時，都會將有關驗證過程的信息洩漏到模型中。

重複幾次，這是無害的；在多次迭代中系統地完成，

最終會導致您的模型過度擬合驗證過程（即使沒有模型直接在任何驗證數據上進行訓練）。

這使得評估過程不太可靠。

一旦您開發出令人滿意的模型配置，您就可以在所有可用數據

（訓練和驗證）上訓練您的最終生產模型，並在測試集上最後一次對其進行評估。如果事實證明測試集上的性能明顯低於驗證數據上測量的性能，這可能意味著您的驗證過程根本不可靠，或者您開始調整模型參數的同時過度擬合驗證數據。在這種情況下，您可能希望切換到更可靠的評估協議（例如迭代 K 折驗證）。

6.3.2發布模型

將模型部署為 REST API

這可能是將模型變成產品的常用方法：在服務器或云實例上安裝TensorFlow，並通過 REST API 查詢模型的預測。您可以使用諸如 Flask

（或任何其他 Python Web 開發庫）之類的東西構建自己的服務應用程序，或者使用 TensorFlow 自己的庫將模型作為 API 提供，稱為TensorFlow Serving ([www.tensorflow.org/tfx/guide/serving)。使用TensorFlow](http://www.tensorflow.org/tfx/guide/serving)。使用TensorFlow) Serving，您可以在幾分鐘內部署 Keras 模型。

您應該在以下情況下使用此部署設置

將使用模型預測的應用程序將可以可靠地訪問互聯網（顯然）。例如，如果您的應用程序是移動應用程序，從遠程 API 提供預測意味著該應用程序將無法在飛行模式或低連接性環境中使用。

該應用程序沒有嚴格的延遲要求：請求、推理

和應答往返通常需要大約 500 毫秒。

發送用於推理的輸入數據不是高度敏感的：數據需要在

服務器以解密的形式，因為模型需要看到它（但請注意，您應該對 HTTP 請求和應答使用 SSL 加密）。

例如，圖像搜索引擎項目、音樂推薦系統、信用卡欺詐檢測項目和衛星圖像項目都非常適合通過 REST API 提供服務。

將模型部署為 REST API 時的一個重要問題是，您是要自己託管代碼，還是要使用完全託管的第三方雲服務。

例如，Google 產品 Cloud AI Platform 可讓您簡單地將 TensorFlow 模型上傳到 Google Cloud Storage (GCS)，並為您提供 API 端點來查詢 它。它處理許多實際細節，例如批量預測、負載平衡和縮放。

您的模型可以做得足夠小，使其可以在目標設備的內存和功率

限制下運行。您可以使用 TensorFlow 模型優化工具包來幫助解決此問題(www.tensorflow.org/model\_optimization)。

獲得盡可能高的準確性對您的任務來說並不是關鍵任務。運行時效率和準確性之間總是需要權衡取捨，因此內存和功率 限制通常要求您交付的模型不如可以在大型 GPU 上運行的最佳模型。

要在智能手機或嵌入式設備上部署 Keras 模型，您的首選解決方案是 TensorFlow Lite ([www.tensorflow.org/lite)。它是一個](http://www.tensorflow.org/lite)。它是一個)

高效的設備深度學習推理框架，可在 Android 和 iOS 智能手機以及基於 ARM64 的計算機、Raspberry Pi 或某些微控制器上運行。它包括一個轉換器，可以直接將您的 Keras 模型轉換為 TensorFlow Lite 格式。

概括

當你進行一個新的機器學習項目時，首先要定義手頭的問題：

* 了解更廣泛的背景
* 你打算做什麼 最終目標是什麼，限制是什麼？
* 收集和註釋數據集；確保您深入了解您的數據。
* 選擇你將如何衡量你的問題的成功 你將監控你的驗證數據的哪些指標？

一旦你理解了問題並且你有一個合適的數據集，開發一個模型：