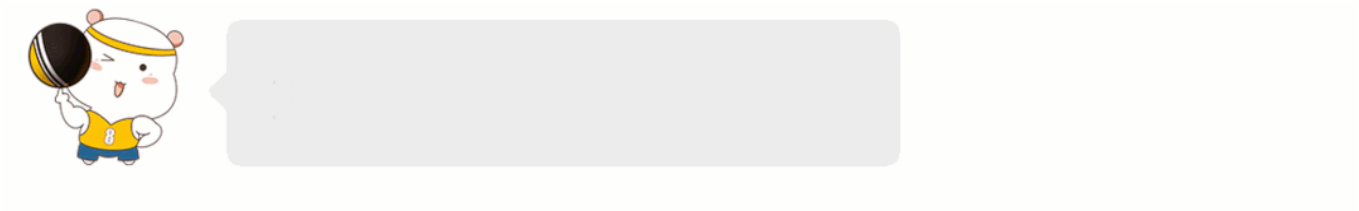
機器學習面試的12個基礎問題

數據STUDIO 2022-08-17 11:30 發表於四川



畢業季找工作了？如果想應聘機器學習工程師崗位，你可能會遇到技術面試，這是面試官掂量你對技術的真正理解的時候，所以還是相當重要的。近日，JP Tech 發表了一篇文章，介紹了他們面試新人時可能會提出的12 個面試問題。問題很基礎，但卻值得一看。

這些問題是我在面試AI 工程師崗位時常問到的問題。事實上，並非所有面試都需要用到所有這些問題，因為這取決於面試者的經驗以及之前做過的項目。經過很多面試（尤其是與學生的面試）之後，我收集了12 個深度學習領域的面試問題。我將在本文中將其分享給你。

問題1

闡述批歸一化的意義

這是一個非常好的問題，因為這涵蓋了面試者在操作神經網絡模型時所需知道的大部分知識。你的回答方式可以不同，但都需要說明以下主要思想：

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

算法1：批歸一化變換，在一個mini-batch 上應用於激活x。

批歸一化是一種用於訓練神經網絡模型的有效方法。這種方法的目標是對特徵進行歸一化處理（使每層網絡的輸出都經過激活），得到標準差為1 的零均值狀態。所以其相反的現像是非零均值。這將如何影響模型的訓練：

首先，這可以被理解成非零均值是數據不圍繞0 值分佈的現象，而是數據的大多數值大於0 或小於0。結合高方差問題，數據會變得非常大或非常小。在訓練層數很多的神經網絡時，這個問題很常見。如果特徵不是分佈在穩定的區間（從小到大的值）裡，那麼就會對網絡的優化過程產生影響。我們都知道，優化神經網絡將需要用到導數計算。

假設一個簡單的層計算公式y = (Wx + b)，y 在W 上的導數就是這樣：dy=dWx。因此，x 的值會直接影響導數的值（當然，神經網絡模型的梯度概念不會如此之簡單，但理論上，x 會影響導數）。因此，如果x 引入了不穩定的變化，則這個導數要么過大，要么就過小，最終導致學習到的模型不穩定。而這也意味著當使用批歸一化時，我們可以在訓練中使用更高的學習率。

批歸一化可幫助我們避免x 的值在經過非線性激活函數之後陷入飽和的現象。也就是說，批歸一化能夠確保激活都不會過高或過低。這有助於權重學習——如果不使用這一方案，某些權重可能永遠不會學習。這還能幫助我們降低對參數的初始值的依賴。

批歸一化也可用作正則化（regularization）的一種形式，有助於實現過擬合的最小化。使用批歸一化時，我們無需再使用過多的dropout；這是很有助益的，因為我們無需擔心再執行dropout時丟失太多信息。但是，仍然建議組合使用這兩種技術。

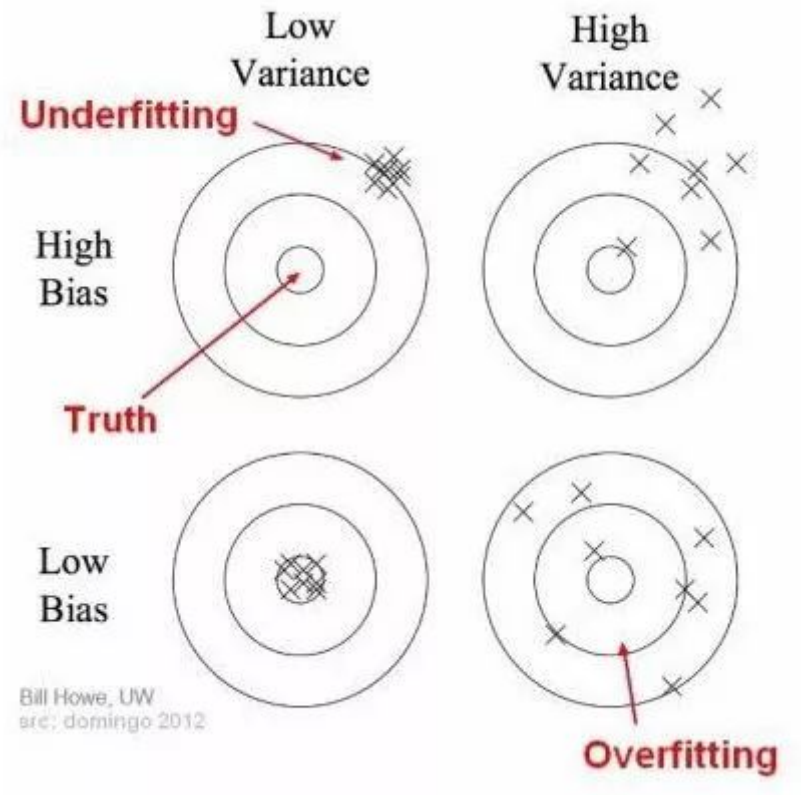
問題2

闡述偏置和方差的概念以及它們之間的權衡關係

偏置（bias）是什麼？這很好理解，偏置是當前模型的平均預測結果與我們需要預測的實際結果之間的差異。當模型的偏置較高時，說明其不夠關注訓練數據。這會使得模型過於簡單，無法在訓練和測試上同時實現優良的準確度。這個現像也被稱為「欠擬合」。

方差（variance）可以簡單理解為是模型輸出在一個數據點上的分佈（或聚類）。方差越大，模型越有可能更密切關注訓練數據，而無法提供在從未見過的數據上的泛化能力。由此造成的結果是，模型可在訓練數據集上取得非常好的結果，但在測試數據集上的表現卻非常差。這個現像被稱為過擬合。

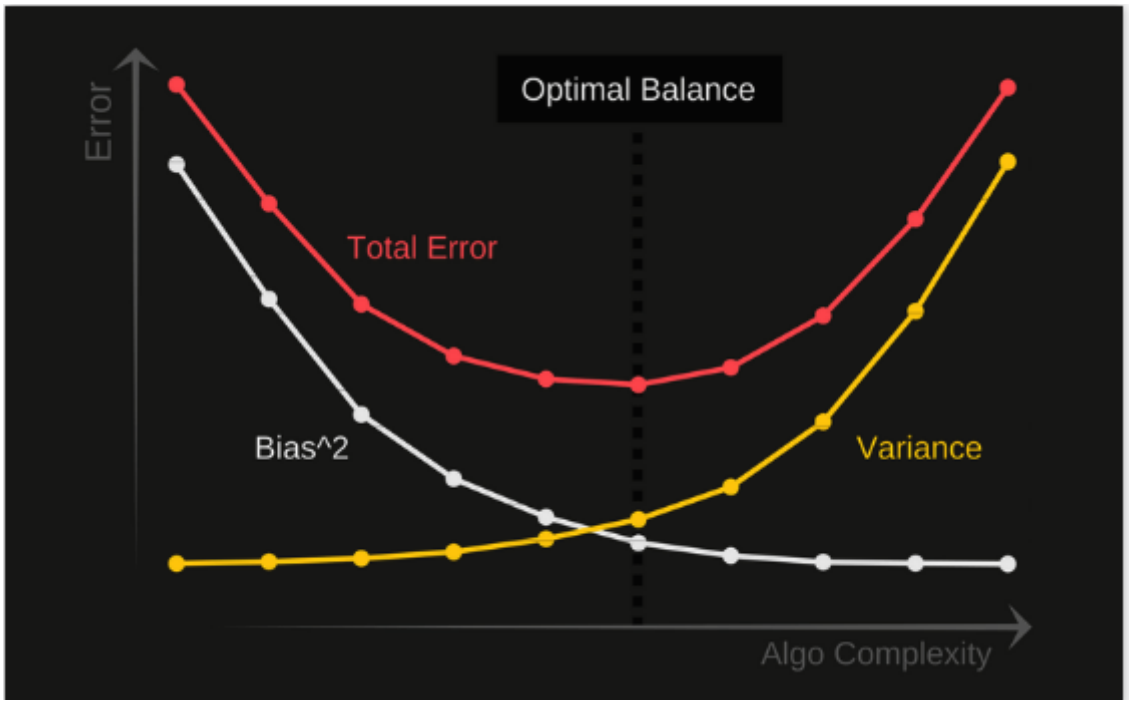
這兩個概念的關係可通過下圖說明：



上圖中，圓圈中心是能夠完美預測精準值的模型。事實上，你永遠無法找到這樣好的模型。隨著我們離圓圈中心越來越遠，模型的預測也越來越差。

我們可以改變模型，使得我們可以增大模型猜測的數量，使其盡可能多地落在圓圈中心。偏置和方差之間需要保持平衡。如果我們的模型過於簡單，有非常少的參數，那麼它就可能有較高的偏置和較低的方差。

另一方面，如果我們的模型有大量參數，則其將有較高的方差和較低的偏置。這是我們在設計算法時計算模型複雜度的基礎。

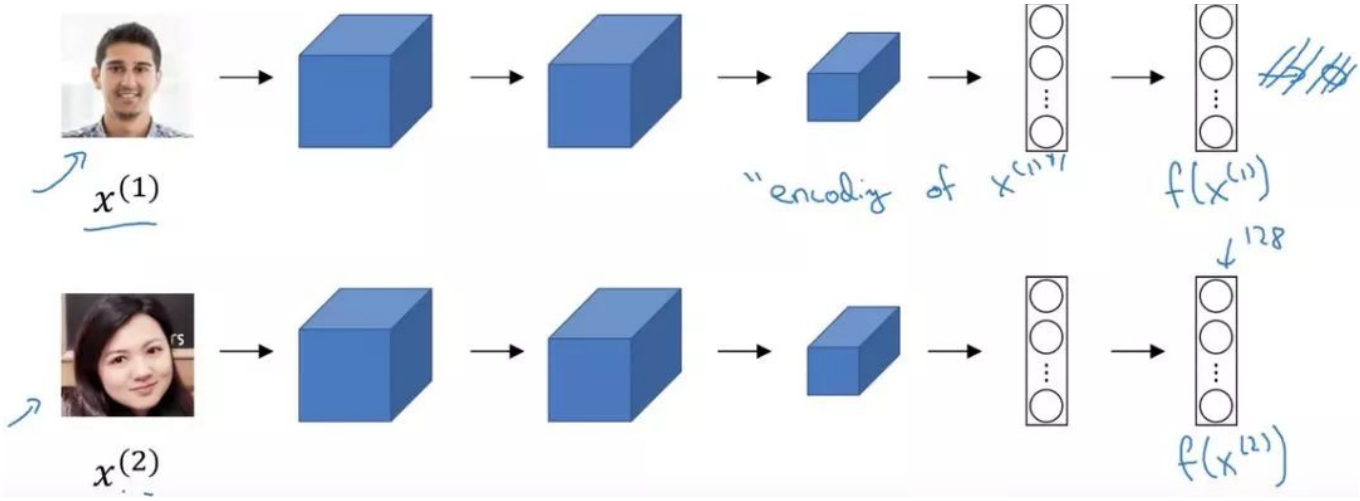


問題3

假設深度學習模型已經找到了1000 萬個人臉向量，如何通過查詢以最快速度找到一張新人臉?

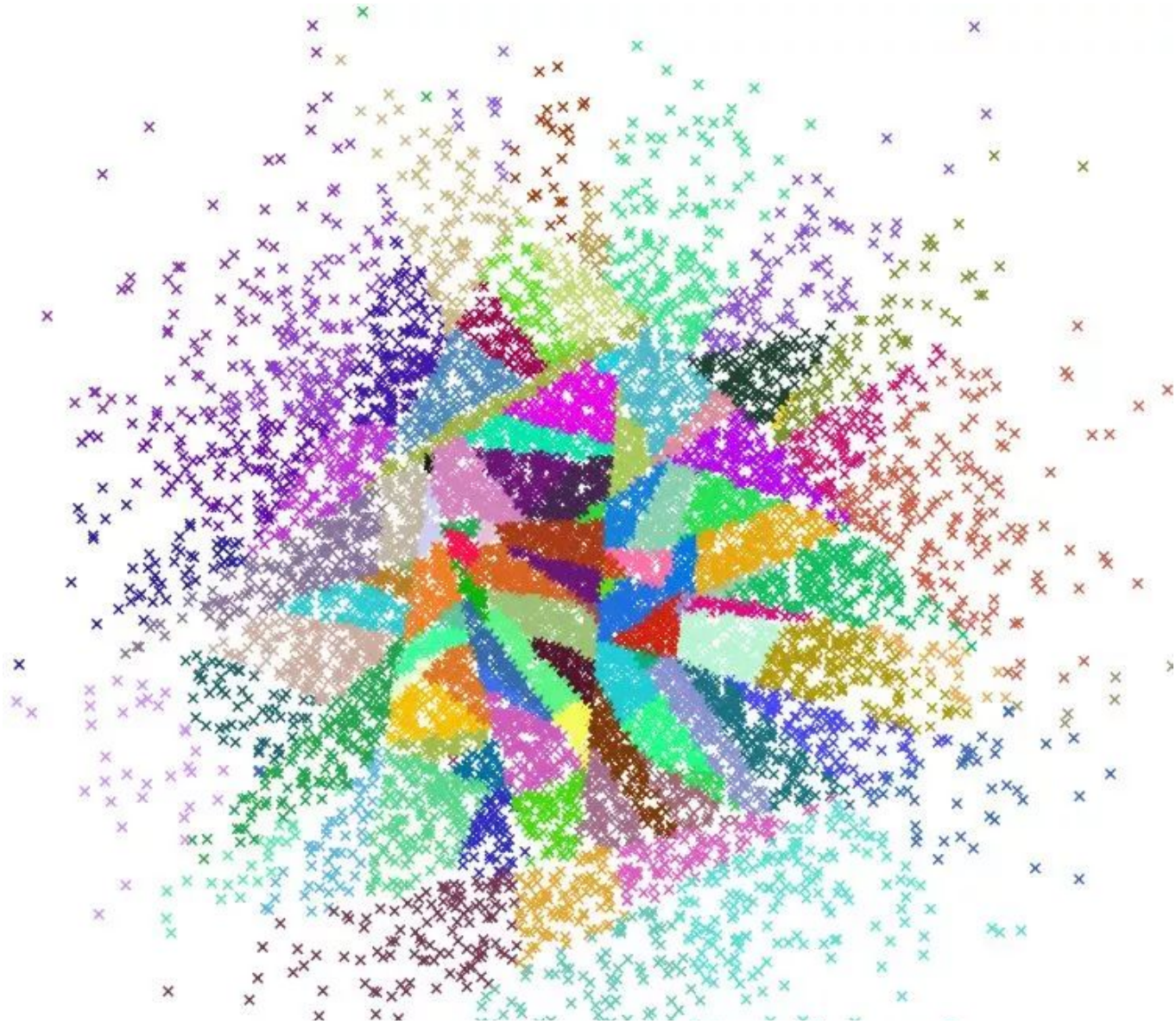
這個問題涉及到深度學習算法的實際應用，關鍵點在於**索引數據**的方法。這是將One Shot Learning應用於人臉識別的最後一步，但這也是最重要的步驟，讓該應用易於實際部署。

基本上來說，對於這個問題，你首先應該通過One Shot Learning 給出人臉識別方法的整體概況。這可以簡單地理解成將每張臉轉換成一個向量，然後識別新的人臉是尋找最接近（最相似）於輸入人臉的向量。通常來說，人們會使用有三元組損失（triplet loss）的定制損失函數的深度學習模型來完成這一任務。



但是，如果有文章開頭那樣的圖像數量增長，那麼在每次識別中都計算與1000 萬個向量的距離可不是個聰明的解決方案，這會使得系統的速度非常慢。我們需要思考在真實向量空間上索引數據的方法，以便讓查詢更加便捷。

這些方法的主要思想是將數據劃分成簡單的結構，以便查詢新數據（可能類似於樹結構）。當有新數據時，在樹中查詢有助於快速找到距離最近的向量。



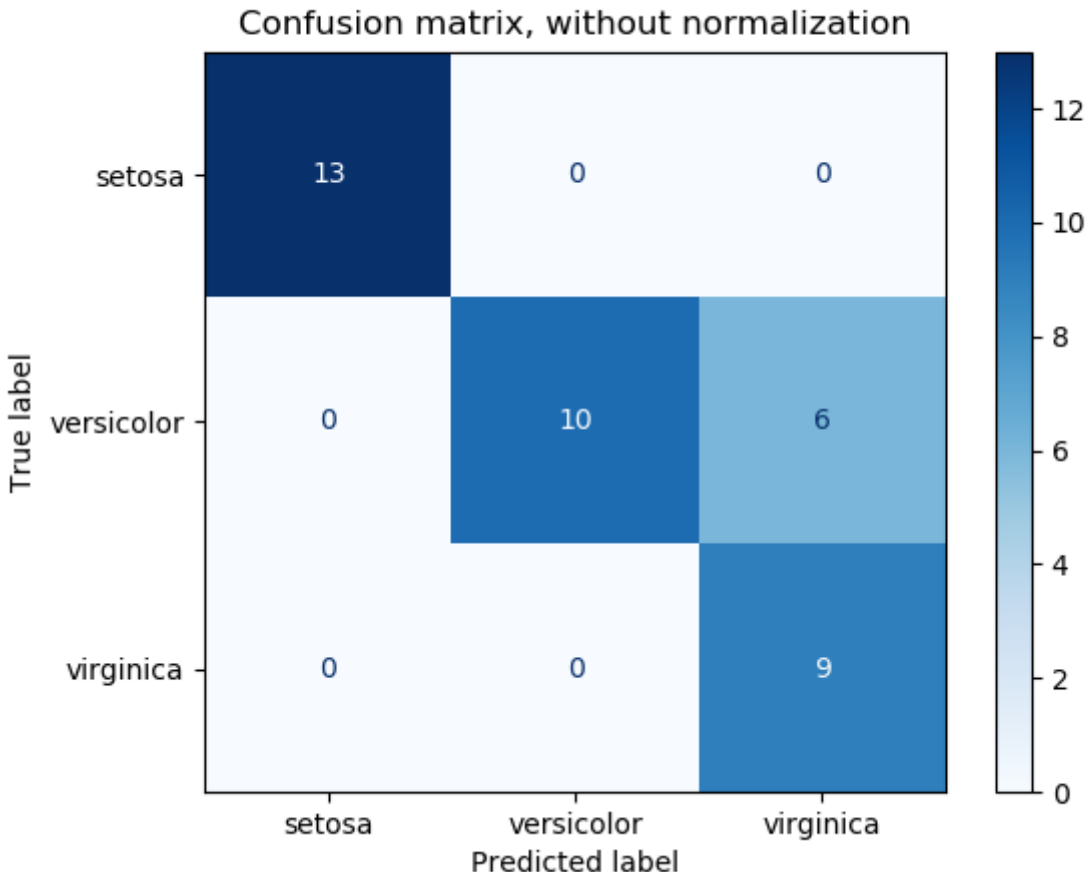
有一些可以用於這一目的的方法，比如局部敏感哈希（LSH）、Approximate Nearest Neighbors Oh Yeah——Annoy Indexing、Faiss等。

問題4

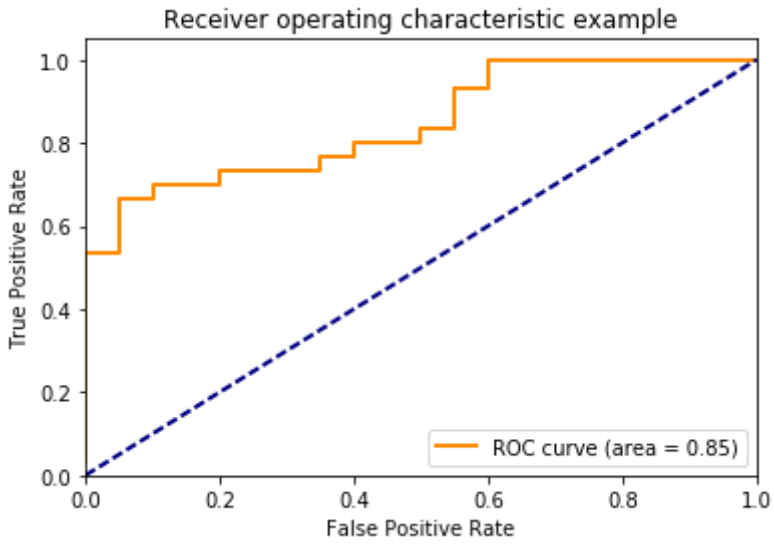
對於分類問題，準確度指數完全可靠嗎？你通常使用哪些指標來評估你的模型？

針對分類問題的評估方法有很多。準確度是一種很簡單的指標，也就是用正確的預測數據除以總的數據。這聽起來很合理，但現實情況是，這種度量方式對不平衡的數據問題來說並不夠顯著。假設我們正在構建用於預測網絡攻擊的預測模型（假設攻擊請求大約佔請求總數的1/100000）。

如果該模型預測所有請求都是正常的，那麼其準確率也高達99.9999%，但在這個分類模型中，這個數字通常是不可靠的。上面的準確度計算得到的結果通常是被正確預測的數據的百分比，但沒有詳細說明每個類別的分類細節。相反，我們可以使用混淆矩陣。基本上來說，混淆矩陣展示了數據點實際屬於的類別，以及模型預測的類別。其形式如下：



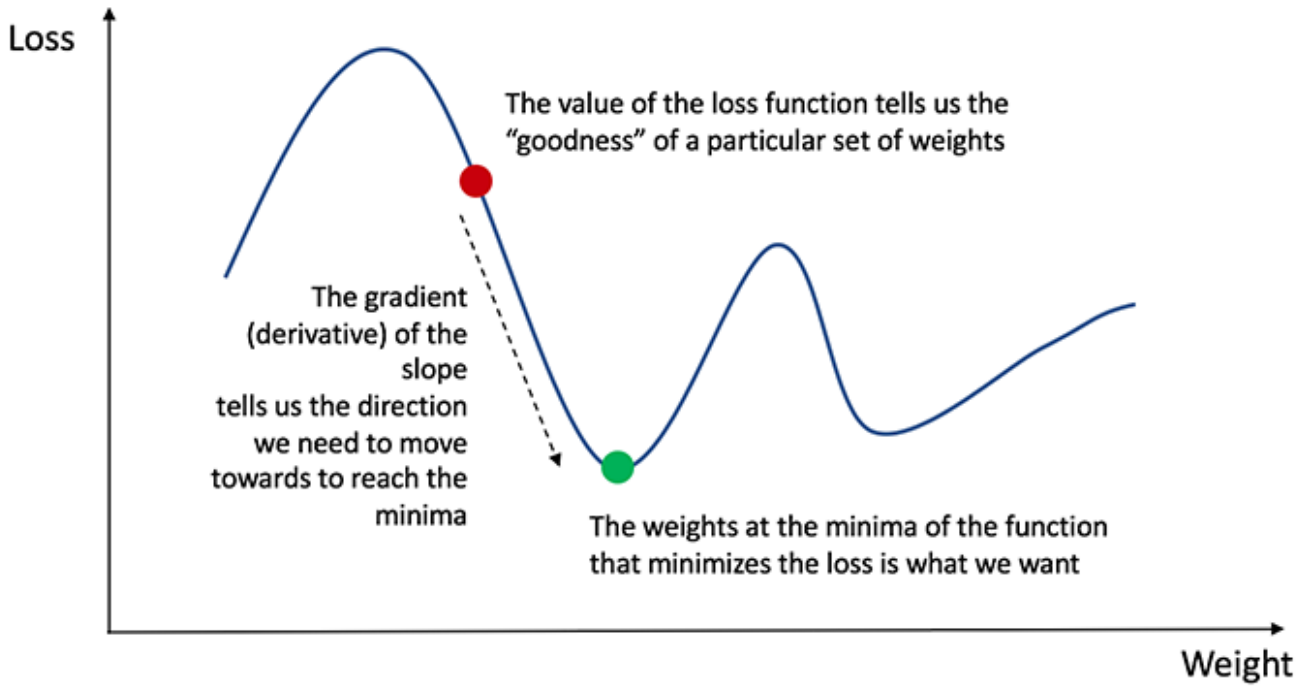
除了表達真正例和假正例指標對應於定義了該分類的每個閾值的變化之外，我們還有名為受試者工作特徵（ROC）的圖表。基於ROC，我們可以知道該模型是否有效。



理想的ROC 越接近左上角的橙色線（即真正例較高，假正例較低），結果就越好。

問題5

你怎麼理解反向傳播？請解釋動作（action）的機制。



這個問題的目標是測試參加面試的人是否理解神經網絡的工作方式。你需要說明以下幾點：

前向過程（前向計算）是幫助模型計算每層的權重的過程，所得到的計算會得到一個結果yp。這時候會計算損失函數的值；損失函數的這個值能體現模型的優劣程度。如果這個損失函數不夠好，我們就需要找到一種能夠降低這個損失函數的值的方法。神經網絡的訓練目標實際上就是最小化某個損失函數。損失函數L(yp,yt) 表示yp 模型的輸出值與yt 數據標籤的實際值之間的差異程度。

為了降低損失函數的值，我們需要使用導數。反向傳播能幫助我們計算網絡每一層的導數。基於每一層上導數的值，優化器（Adam、SGD、AdaDelta 等）可通過梯度下降來更新網絡的權重。

反向傳播會使用鍊式法則機製或導數函數，從最後一層到第一層計算每一層的梯度值。

問題6

激活函數有什麼含義？激活函數的飽和點是什麼？

1. 激活函數的含義

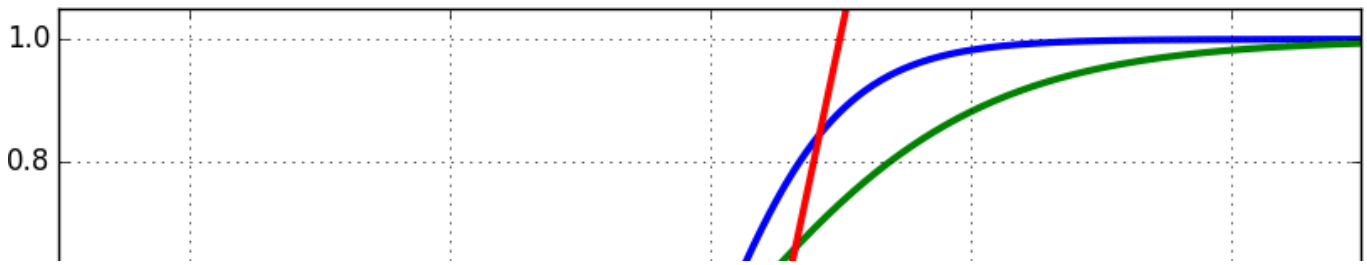
激活函数的目的是突破神经网络的线性性质。我们可以将这些函数简单理解成是一种过滤器，作用是决定信息是否可以通过神经元。在神经网络训练期间，激活函数在调整导数斜率方面具有非常重要的作用。

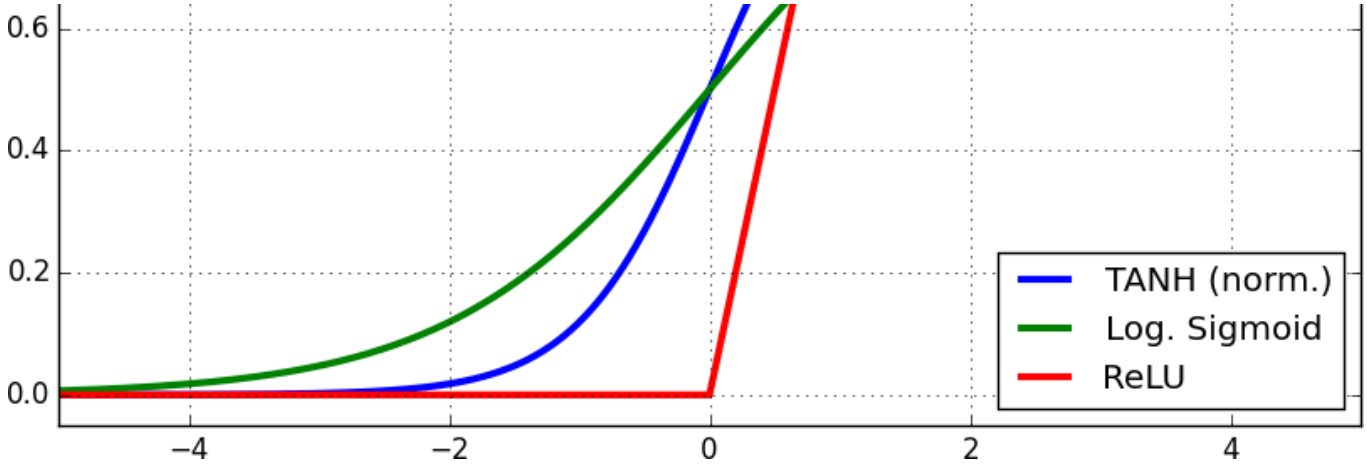
相比于使用线性函数，使用非线性激活函数能让神经网络学习更复杂的函数表征；但为了有效地使用它们，我们需要理解这些非线性函数的性质。大多数激活函数都是连续可微的函数。

这些函数是连续函数，也就是说如果输入有较小的可微分的变化（在其定义域中的每个点上都有导数），那么输出也会有较小的变化。当然，如前面提到的那样，导数的计算是非常重要的，而且决定了我们的神经元是否可以训练。值得提及的几种激活函数有 Sigmoid、Softmax 和 ReLU。

2. 激活函数的饱和范围

Tanh、Sigmoid 和 ReLU 函数等非线性激活全都有饱和区间。





很容易理解，激活函数的饱和范围就是当输入值变化时输出值不再变化的区间。这个变化区间存在两个问题。

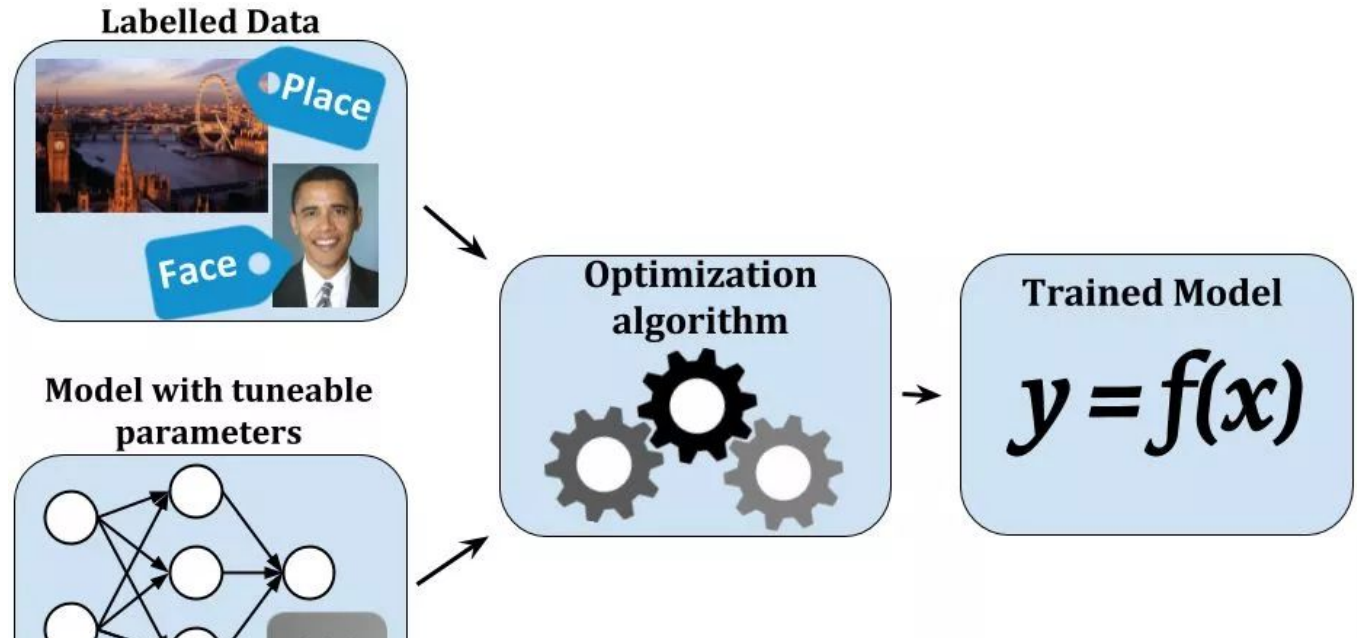
第一个问题是在神经网络的前向方向上，落在激活函数的饱和范围内的层的值将会逐渐得到许多同样的输出值。这会导致整个模型出现同样的数据流。这个现象被称为协方差偏移（covariance shifting）。

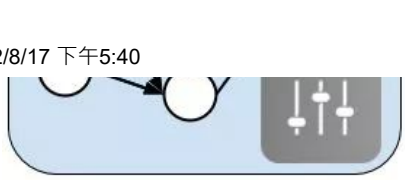
第二个问题是在反向方向上，饱和范围内的导数为零，由此导致网络几乎无法再学习到任何东西。这就是我们在批归一化问题中提到的要将值的范围设定为零均值的原因。

问题7

模型的超参数是什么？超参数与参数有何不同?

1. 模型参数是什么？





先稍微回顾一下机器学习的本质，要做机器学习，我们需要有一个数据集。没有数据我们怎么学习呢？一旦有了数据，机器需要找到数据之间的关联。

假设我们的数据是温度和湿度等天气信息，我们希望机器执行的任务是找到这些因素与我们的爱人是否生气之间的关联。这听起来似乎并无关联，但机器学习的待办事项有时候确实很可笑。现在，我们用变量 y 表示我们的爱人是否生气，变量 x_1、x_2、x_3……表示天气元素。我们用下面的函数 f(x) 表示这些变量之间的关系：

$$y = f(x) = w_1.x_1 + w_2.x_2 + w_3.x_3$$

看到系数 w_1、w_2、w_3 了吗？这就代表了数据和结果之间的关系，这就是所谓的模型参数。因此，我们可以这样定义「模型参数」：

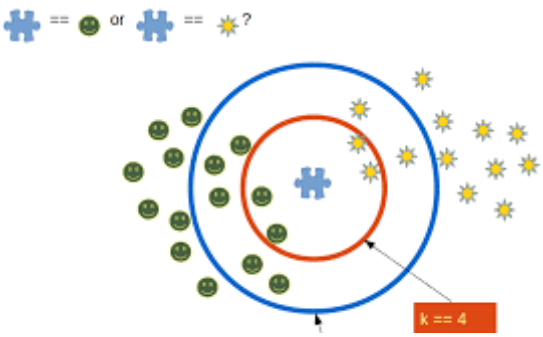
模型参数是模型基于训练数据生成的值，有助于展示数据中数据量之间的关系。

所以当我们说要为某问题找到最佳的模型时，我们的意思是要基于已有的数据集为该问题找到最合适的模型参数。模型参数有如下特性：

- 可用于预测新数据；
- 能展现我们使用的模型的能力，通常通过准确度等指标表示；
- 是直接从训练数据集学习到的；
- 不是由人类人工设置的。

模型参数也有不同的形式，比如在神经网络中是权重、在支持向量机中是支持向量、在线性回归和 logistic 回归算法中是系数。

2. 什么是模型超参数？



k == 11

可能有人认为模型超参数就是或者像是模型参数，但事实并非如此。实际上这两个概念是完全不同的。模型参数是从训练数据集建模的，而模型超参数却完全不是这样，其完全位于模型之外而且不依赖于训练数据。所以模型超参数的作用是什么？实际上它们有以下任务：

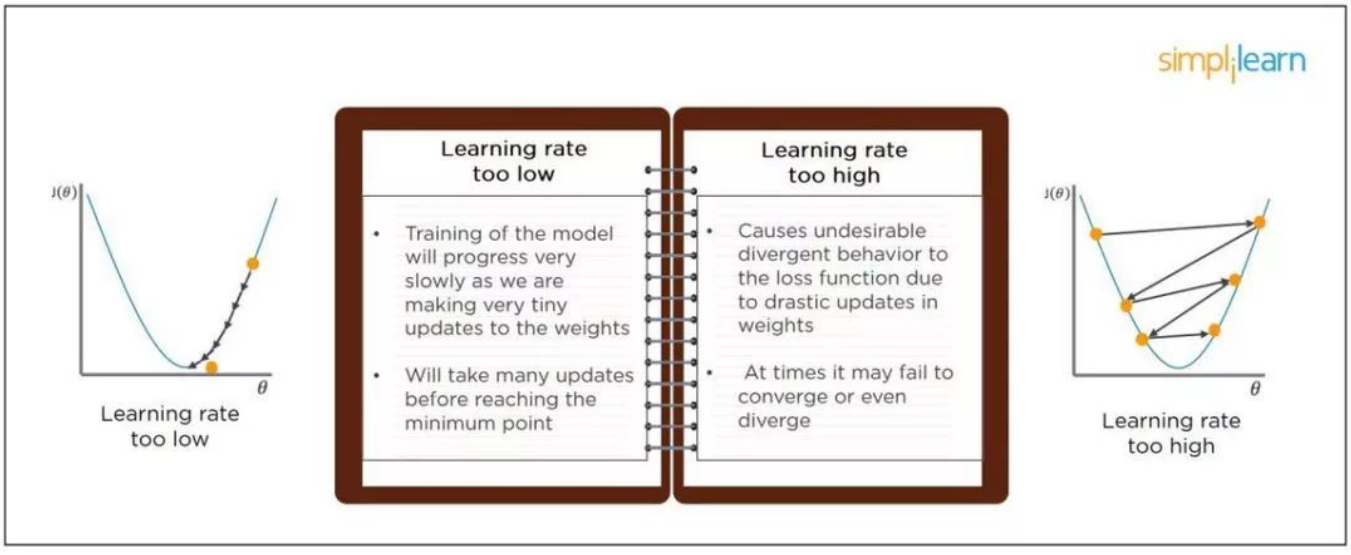
- 在训练过程中使用，帮助模型寻找最合适的参数；
- 通常是在模型设计时由人工选择的；
- 可基于几种启发式策略来定义。

对于某个具体问题，我们完全不知道最佳的超参数模型是怎样的。因此，实际上我们需要使用某些技术（比如网格搜索）来估计这些值的最佳范围（比如，k 最近邻模型中的 k 系数）。下面是模型超参数的一些示例：

- 训练人工神经网络时的学习率指数；
- 训练支持向量机时的 C 和 σ 参数；
- k 最近邻模型中的 k 系数。

问题8

当学习率过高或过低时会怎样？



当模型的学习率过低时，模型的训练速度会变得非常慢，因为其每次对权重的更新会变得非常小。模型将需要大量更新才能到达局部最优点。

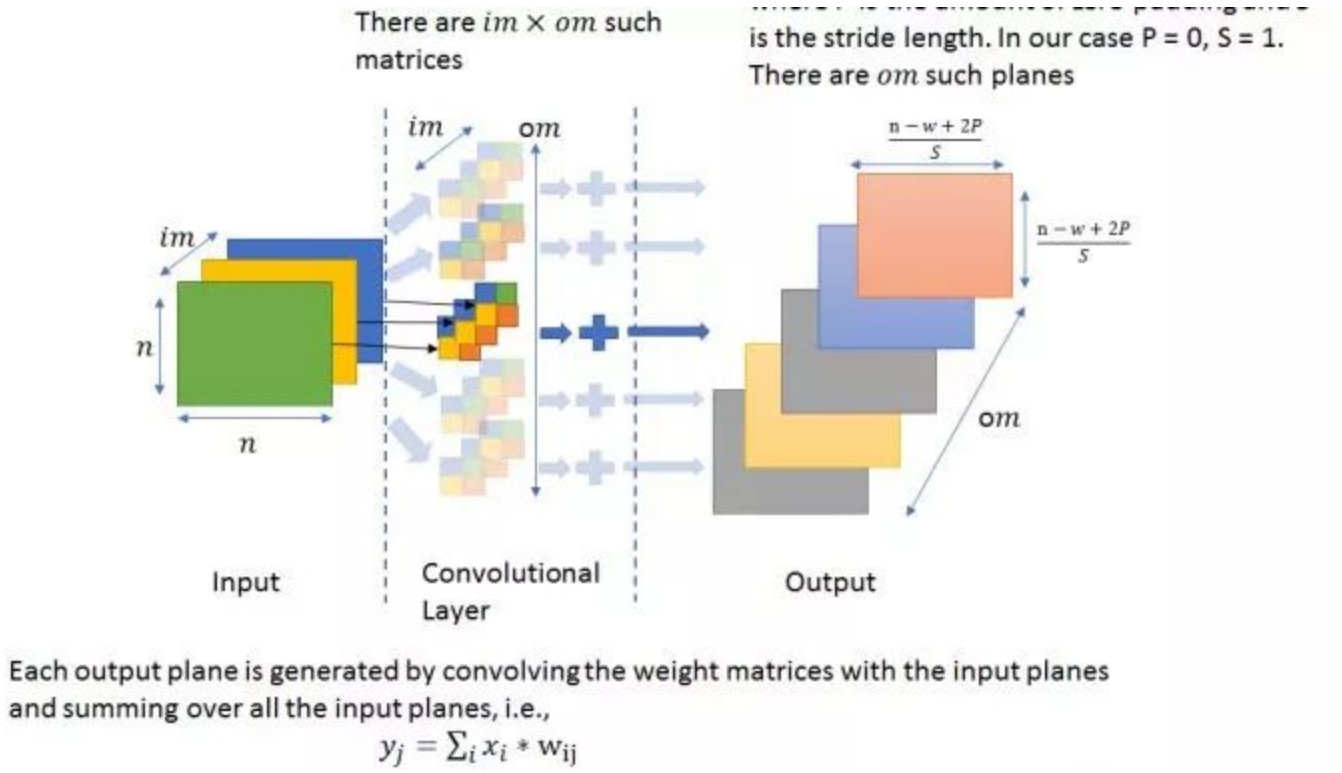
如果学习率过高，模型很可能无法收敛，因为权重的更新过大。在加权的步骤中，模型有可能无法实现局部优化，然后使模型难以更新到最优点（因为每步更新都跳得过远，导致模型在局部最优点

附近摇摆）。

问题9

当输入图像的尺寸加倍时，CNN 参数的数量会增加多少倍？为什么？

对于参加面试的人来说，这个问题很有误导性，因为大部分人思考这个问题的方向都是 CNN 的参数数量会增加多少倍。但是，我们看看 CNN 的架构：



可以看到，CNN 模型的参数数量取决于过滤器的数量和大小，而非输入图像。因此，将输入图像的尺寸加倍不会改变模型的参数数量。

问题10

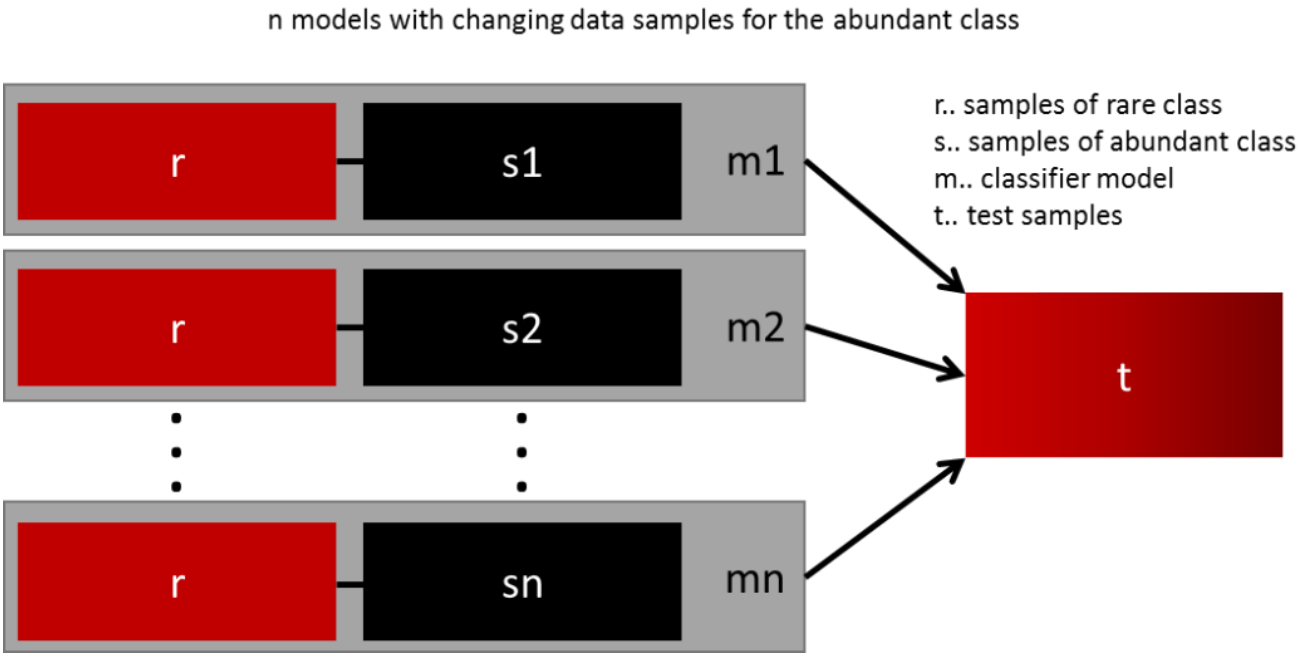
处理数据不平衡问题的方法有哪些？

这个问题检验的是面试者是否知道处理有真实数据的问题的方法。通常来说，实际数据和样本数据（无需调整的标准数据集）在性质和数据量上都有很大的不同。使用真实数据集时，数据有可能是不平衡的，也就是说不同类别的数据不平衡。针对这个问题，我们可以考虑使用以下技术：

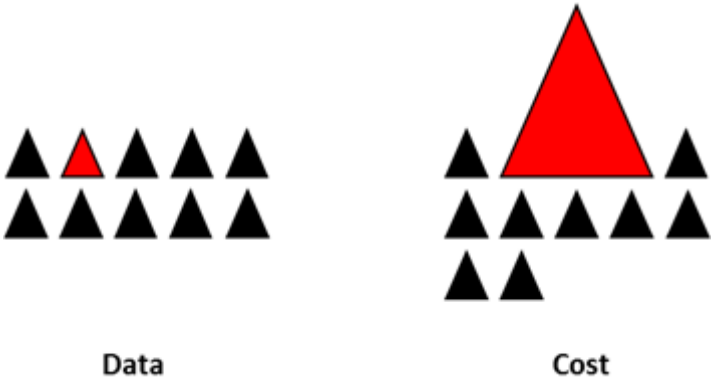
为模型的评估选择适当的指标：当使用的数据集不平衡时，使用准确度来进行评估是很不合适的（前面已经提到过），而应该选择精确度、召回率、F1 分数、AUC 等评估指标。

对训练数据集进行重新采样：除了使用不同的评估指标外，人们还可以通过某些技术来获得不同的数据集。基于不平衡的数据集创建平衡的数据集的方法有两种：欠采样和过采样，具体技术包括重复、自举或 SMOTE（合成少数过采样技术）。

集成多个不同模型：通过创建更多数据来实现模型的通用性在实践中是不可取的。举个例子，假设你有两个类别：一个有 1000 个数据样本的罕见类别以及一个有 10000 个数据样本的常见类别。我们可以不必努力为罕见类别寻找 9000 个数据样本来进行模型训练，而是可以采用一种 10 个模型的训练方案。其中每个模型都使用 1000 个罕见数据样本和 1000 个常见数据样本进行训练。然后使用集成技术得到最佳结果。



重新设计模型——成本函数：在成本函数中使用惩罚技术来严厉惩罚数据丰富的类别，以帮助模型自身更好地学习罕见类别的数据。这能使损失函数的值更全面地覆盖所有类别。



问题11

在训练深度学习模型时，epoch、batch（批）和 iteration（迭代）这些概念都是什么意思?

这些是训练神经网络时非常基本的概念，但实际上很多面试者在区分这些概念时常常搞混淆。具体来说，你应该这样回答：

- epoch：代表在整个数据集上的一次迭代（所有一切都包含在训练模型中）；
- batch：是指当我们无法一次性将整个数据集输入神经网络时，将数据集分割成的一些更小的数据集批次；
- iteration：是指运行一个 epoch 所需的 batch 数。举个例子，如果我们的数据集包含 10000 张图像，批大小（batch_size）是 200，则一个 epoch 就包含 50 次迭代（10000 除以 200）。

问题12

数据生成器的概念是什么？使用数据生成器需要什么?

生成函数在编程中也非常重要。数据生成函数可帮助我们在每个训练 batch 中生成能直接拟合模型的数据。

使用生成函数在训练大数据时大有助益。因此数据集并不是需要全部都载入 RAM，这是浪费内存；此外，如果数据集过大，还可能导致内存溢出，对输入数据的处理时间也会变得更长。

总结

上面就是我常在面试过程中向参加面试的人提出的 12 个有关深度学习的面试问题。但是，根据每个面试者的情况不同，提问的方式可以也会各不相同，另外也会有其它一些根据面试者的经历而提出的问题。

尽管这篇文章只涉及技术问题，但也是与面试相关的。在我个人看来，态度是面试成功的一半。所以除了让你自己积累知识和技能之外，一定要用真正、进取又谦虚的态度展现你自己，这样能让你在对话中取得很大的成功。

> 作者：JP Tech等
> 机器之心编译
> 来源：https://medium.com/

【Spark】Spark 高頻面試題英文版(1)

BigDataNotes