1.简介

逻辑回归是面试当中非常喜欢问到的一个机器学习算法，因为表面上看逻辑回归形式上很简单，很好掌握，但是一问起来就容易懵逼。所以在面试的时候给大家的第一个建议不要说自己精通逻辑回归，非常容易被问倒，从而减分。下面总结了一些平常我在作为面试官面试别人和被别人面试的时候，经常遇到的一些问题。

2.正式介绍

如何凸显你是一个对逻辑回归已经非常了解的人呢。那就是用一句话概括它！**逻辑回归假设数据服从伯努利分布,通过极大化似然函数的方法，运用梯度下降来求解参数，来达到将数据二分类的目的。**

这里面其实包含了 5 个点 1：逻辑回归的假设，2：逻辑回归的损失函数，3：逻辑回归的求解方法，4：逻辑回归的目的，5:逻辑回归如何分类。这些问题是考核你对逻辑回归的基本了解。

- **逻辑回归的基本假设**
  - 任何的模型都是有自己的假设，在这个假设下模型才是适用的。逻辑回归的**第一个**基本假设是**假设数据服从伯努利分布。**伯努利分布有一个简单的例子是抛硬币，抛中为正面的概率是,抛中为负面的概率是.在逻辑回归这个模型里面是假设 为样本为正的概率，为样本为负的概率。那么整个模型可以描述为
  - 逻辑回归的第二个假设是假设样本为正的概率是
  - 所以逻辑回归的最终形式
- **逻辑回归的损失函数**
  - 逻辑回归的损失函数是它的极大似然函数
- **逻辑回归的求解方法**
  - 由于该极大似然函数无法直接求解，我们一般通过对该函数进行梯度下降来不断逼急最优解。在这个地方其实会有个加分的项，考察你对其他优化方法的了解。因为就梯度下降本身来看的话就有随机梯度下降，批梯度下降，small batch 梯度下降三种方式，面试官可能会问这三种方式的优劣以及如何选择最合适的梯度下降方式。
    - 简单来说 批梯度下降会获得全局最优解，缺点是在更新每个参数的时候需要遍历所有的数据，计算量会很大，并且会有很多的冗余计算，导致的结果是当数据量大的时候，每个参数的更新都会很慢。

- 随机梯度下降是以高方差频繁更新，优点是使得 sgd 会跳到新的和潜在更好的局部最优解，缺点是使得收敛到局部最优解的过程更加的复杂。
- 小批量梯度下降结合了 sgd 和 batch gd 的优点，每次更新的时候使用 n 个样本。减少了参数更新的次数，可以达到更加稳定收敛结果，一般在深度学习当中我们采用这种方法。

-

- 其实这里还有一个隐藏的更加深的加分项，看你了不了解诸如 Adam，动量法等优化方法。因为上述方法其实还有两个致命的问题。

- 第一个是如何对模型选择合适的学习率。自始至终保持同样的学习率其实不太合适。因为一开始参数刚刚开始学习的时候，此时的参数和最优解隔的比较远，需要保持一个较大的学习率尽快逼近最优解。但是学习到后面的时候，参数和最优解已经隔的比较近了，你还保持最初的学习率，容易越过最优点，在最优点附近来回振荡，通俗一点说，就很容易学过头了，跑偏了。
- 第二个是如何对参数选择合适的学习率。在实践中，对每个参数都保持的同样的学习率也是很不合理的。有些参数更新频繁，那么学习率可以适当小一点。有些参数更新缓慢，那么学习率就应该大一点。这里我们不展开，有空我会专门出一个专题介绍。

- **逻辑回归的目的**

- 该函数的目的便是将数据二分类，提高准确率。

- **逻辑回归如何分类**

- 逻辑回归作为一个回归(也就是 y 值是连续的)，如何应用到分类上去呢。y 值确实是一个连续的变量。逻辑回归的做法是划定一个阈值，y 值大于这个阈值的是一类，y 值小于这个阈值的是另外一类。阈值具体如何调整根据实际情况选择。一般会选择 0.5 做为阈值来划分。

3.对逻辑回归的进一步提问

逻辑回归虽然从形式上非常的简单，但是其内涵是非常的丰富。有很多问题是可以进行思考的

- **逻辑回归的损失函数为什么要使用极大似然函数作为损失函数？**

- 损失函数一般有四种，平方损失函数，对数损失函数，HingeLoss0-1 损失函数，绝对值损失函数。将极大似然函数取对数以后等同于对数损失函数。在逻辑回归这个模型下，对数损失函数的训练求解参数的速度是比较快的。至于原因大家可以求出这个式子的梯度更新

这个式子的更新速度只和，相关。和 sigmod 函数本身的梯度是无关的。这样更新的速度是可以自始至终都比较的稳定。

- 为什么不选平方损失函数的呢？其一是因为如果你使用平方损失函数，你会发现梯度更新的速度和 sigmod 函数本身的梯度是很相关的。sigmod 函数在它在定义域内的梯度都不大于 0.25。这样训练会非常的慢。

- **逻辑回归在训练的过程当中，如果有很多的特征高度相关或者说有一个特征重复了 100 遍，会造成怎样的影响？**
- 先说结论，如果在损失函数最终收敛的情况下，其实就算有很多特征高度相关也不会影响分类器的效果。
- 但是对特征本身来说的话，假设只有一个特征，在不考虑采样的情况下，你现在将它重复 100 遍。训练以后完以后，数据还是这么多，但是这个特征本身重复了 100 遍，实质上将原来的特征分成了 100 份，每一个特征都是原来特征权重值的百分之一。
- 如果在随机采样的情况下，其实训练收敛完以后，还是可以认为这 100 个特征和原来那一个特征扮演的效果一样，只是可能中间很多特征的值正负相消了。
- **为什么我们还是会在训练的过程当中将高度相关的特征去掉？**

- 去掉高度相关的特征会让模型的可解释性更好
- 可以大大提高训练的速度。如果模型当中有很多特征高度相关的话，就算损失函数本身收敛了，但实际上参数是没有收敛的，这样会拉低训练的速度。其次是特征多了，本身就会增大训练的时间。

4.逻辑回归的优缺点总结

面试的时候，别人也经常会问到，你在使用逻辑回归的时候有哪些感受。觉得它有哪些优缺点。

在这里我们总结了逻辑回归应用到工业界当中一些优点：

- 形式简单，模型的可解释性非常好。从特征的权重可以看到不同的特征对最后结果的影响，某个特征的权重值比较高，那么这个特征最后对结果的影响会比较大。
- 模型效果不错。在工程上是可以接受的（作为 baseline），如果特征工程做的好，效果不会太差，并且特征工程可以大家并行开发，大大加快开发的速度。
- 训练速度较快。分类的时候，计算量仅仅只和特征的数目相关。并且逻辑回归的分布式优化 sgd 发展比较成熟，训练的速度可以通过堆机器进一步提高，这样我们可以在短时间内迭代好几个版本的模型。
- 资源占用小,尤其是内存。因为只需要存储各个维度的特征值，。
- 方便输出结果调整。逻辑回归可以很方便的得到最后的分类结果，因为输出的是每个样本的概率分数，我们可以很容易的对这些概率分数进行 cutoff，也就是划分阈值(大于某个阈值的是一类，小于某个阈值的是一类)。

但是逻辑回归本身也有许多的缺点：

- 准确率并不是很高。因为形式非常的简单(非常类似线性模型)，很难去拟合数据的真实分布。
- 很难处理数据不平衡的问题。举个例子：如果我们对于一个正负样本非常不平衡的问题比如正负样本比 10000:1.我们把所有样本都预测为正也能使损失函数的值比较小。但是作为一个分类器，它对正负样本的区分能力不会很好。
- 处理非线性数据较麻烦。逻辑回归在不引入其他方法的情况下，只能处理线性可分的数据，或者进一步说，处理二分类的问题 。
- 逻辑回归本身无法筛选特征。有时候，我们会用 gbdt 来筛选特征，然后再上逻辑回归。