



链滴

2021 年 4 月下旬，百度机器学习 / 数据挖掘 / NLP 算法工程师面试 8 道题

作者: [julyedu](#)

原文链接: <https://ld246.com/article/1623401995599>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h2 id="文末彩蛋-七月在线干货组最新升级的-名企AI面试100题-免费送-">文末彩蛋：七月在线干货组最新升级的《名企 AI 面试 100 题》免费送！ </h2>

<h2 id="问题1-编程题旋转有序数组-查找元素是否存在">问题 1：编程题旋转有序数组，查找元素是否存在</h2>

<p>思路：</p>

<p>1、暴力破解：遍历整个数组，查找元素是否存在；</p>

<p>2、二分查找：旋转后局部数组依然是有序的，所以此时依然可以使用二分查找算法；</p>

<p>参考代码：</p>

<p></p>

<h2 id="问题2-实现余弦相似度计算">问题 2：实现余弦相似度计算</h2>

<p>余弦相似度:用两个向量夹角判断其相似程度;</p>

<p>向量夹角越大，距离越远，最大距离就是两个向量夹角 180°;</p>

<p>向量夹角越小，距离越近，最小距离就是两个向量夹角 0°，完全重合。</p>

<p>所以余弦相似度越大，向量越相似;</p>

<p>计算公式:</p>

<p></p>

<p>求余弦相似度方法：</p>

<p>Numpy：</p>

<p></p>

<p>Pytorch：</p>

<p></p>

<p>Sklearn：</p>

<p></p>

<h2 id="问题3-验证二叉搜索树-BST-">问题 3：验证二叉搜索树(BST)</h2>

<p>二叉搜索树具有如下特征:</p>

<p>节点的左子树只包含小于当前节点的数。节点的右子树只包含大于当前节点的数。</p>

<p>所有左子树和右子树自身必须也是二叉搜索树。</p>

<p>思路：</p>

<p>根据二叉搜索树的特征可知：如果二叉树的左子树不为空，则左子树上所有节点的值均小于它根节点的值;若它的右子树不为空，则右子树上所有节点的值均大于它根节点的值;并且它的左右子树也为二叉搜索树。</p>

<p>可以设计一个递归函数 check(root, max_num, min_num)，函数表示考虑以 root 为根的子树判断子树中所有节点的值是否都在(max_num, min_num)的范围内。如果 root 节点的值 val 不在(max_num, min_num)的范围内说明不满足条件直接返回 False，否则继续递归，检查它的左右子树是否满足，都满足则说明这是一棵二叉搜索树。</p>

<p>注意：</p>

<p>在递归调用左子树时，需要把上界 max_num 改为 root.val;递归调用右子树时，需要把下界 min_num 改为 root.val.</p>

<p>函数递归调用的入口为 check(root, float("inf"), -float("inf")), float("inf")表示一个无穷大的。</p>

<p>参考代码：</p>

<p></p>

<h2 id="问题4-用randomInt-5-实现randomInt-7--只用讲思路">问题 4: 用 randomInt(5)实现 randomInt(7), 只用讲思路</h2>

<p>randomInt(5):等概率生成整数[1, 2,3,4,5]</p>

<p>randomInt(7):等概率生成整数[1, 2, 3,4,5,6,7]</p>

<p>思路: </p>

<p>1、用(randomInt(5)- 1)构造等概率整数数组:[0,1, 2, 3,4]</p>

<p>2、用(randomInt(5) - 1)*5 构造整数数组 : [0,5,10,15,20]</p>

<p>3、上面的两个整数数组可以构造等概率的新数组:[0,1,2,3,4,5,6,7,... , 24];(如果第 2 个数组选择 2 或者 3 倍, 4 倍则无法构造新的等概率数组)</p>

<p>4、选择新数组[0,1,2,3,..., 20]21 个元组即可构造等概率的数组[1,2,3,4,5,6,7]</p>

<p>参考代码: </p>

<p></p>

<h2 id="问题5-编程题-分割链表问题">问题 5: 编程题: 分割链表问题</h2>

<p>分割链表: </p>

<p>给定一个链表的头节点 head 和一个特定值 x, 然后对链表进行分隔, 使得所有小于 x 的节点都现在大于或等于 x 的节点之前。同时保留两个分区中每个节点的初始相对位置。</p>

<p>如下图所示: </p>

<p></p>

<p>思路: </p>

<p>需维护两个链表 small 和 large, small 链表按顺序存储所有小于 x 的节点, large 链表按顺序存储所有大于等于 x 的节点。遍历完原链表后, 我们只要将 small 链表尾节点指向 large 链表的头节点能完成对链表的分隔。</p>

<p>参考代码: </p>

<p></p>

<h2 id="问题6-怎么解决过拟合-怎么做图像增广-">问题 6: 怎么解决过拟合? 怎么做图像增广? </h2>

<p>常见缓解过拟合的方法: </p>

<p>1、降低模型复杂度</p>

<p>2、增加更多的训练数据:使用更大的数据集训练模型</p>

<p>3、数据增强</p>

<p>4、正则化:L1、L2、添加 BN 层</p>

<p>5、添加 Dropout 策略</p>

<p>6、Early Stopping</p>

<p>7、重新清洗数据:把明显异常的数据剔除</p>

<p>8、使用集成学习方法:把多个模型集成在一起, 降低单个模型的过拟合风险</p>

<p>常见的数据增广方法:</p>

<p>1、水平/垂直翻转</p>

<p>2、随机旋转</p>

<p>3、随机缩放</p>

<p>4、随机剪切</p>

<p>5、颜色、对比度增强</p>

<p>6、cutOut</p>

<p>7、CutMix</p>

<p>8、Mixup</p>

<p>9、Mosaic</p>

<p>10、Random Erasing</p>

<h2 id="问题7-梯度下降方法有哪些-">问题 7: 梯度下降方法有哪些? </h2>

<p>梯度下降算法有如下 3 种:</p>

<p>1、随机梯度下降法:SGD</p>

<p>2、批量梯度下降法:BGD</p>

<p>3、min-batch 小批量梯度下降法:MBGD</p>

<h2 id="问题8-sigmoid有哪些特性-激活函数了解多少-">问题 8: sigmoid 有哪些特性? 激活函数解多少? </h2>

<p></p>

<p>Sigmod 函数性质: </p>

<p>1、定义域:(-oo , +oo); </p>

<p>2、值域:(-1 ,1);</p>

<p>3、函数在定义域内为连续光滑函数;</p>

<p>4、处处可导, 导数为: </p>

<p></p>

<p>5、函数的取值在 0 到 1 之间, 在 0.5 处呈中心对称, 且越靠近 x=0 的取值斜率越大。</p>

<p>常见激活函数: </p>

<p>1、Sigmoid</p>

<p>2、Tanh</p>

<p>3、Relu . Leaky Relu、P-Relu (Parametric ReLU)</p>

<p>4、Elu、Gelu</p>

<p>5、Swich</p>

<p>6、Selu</p>

<p>评论有奖: 评论区回复 "100 题" , 免费领取最新升级版《名企 AI 面试 100 题》电子书! </p>