One-Hot 编码

-概念

- One-Hot 编码，又称为一位有效编码，主要是采用 N 位状态寄存器来对 N 个状态进行编码，每个状态都由他独立的寄存器位，并且在任意时候只有一位有效。
- One-Hot 编码是分类变量作为**二进制向量**的表示。

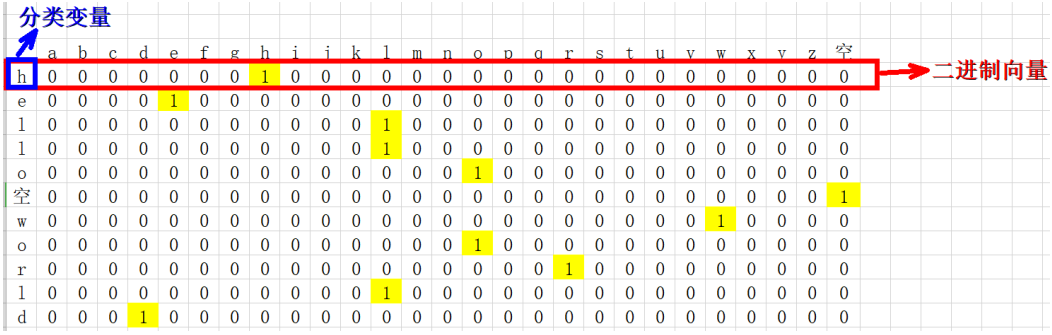
-过程详解

- 例 1：对‘hello world’进行 one-hot 编码
  1. 确定要编码的对象--hello world
  2. 确定分类变量 — h e l l o 空格 w o r l d，共 27 种类（26 个小写字母加空格）
  3. 以上问题就相当于，有 11 个样本，每个样本有 27 个特征，将其转化为二进制向量表示。这里有一个前提，**特征排列的顺序不同，对应的二进制向量亦不同**（比如我把空格放在第一列和 a 放第一列，one-hot 编码结果肯定是不同的）
  4. 因此我们必须要事先约定特征排列的顺序:

  a) 27 种特征首先进行整数编: a-0，b-1，c-2，……，Z-25，空格-26

  b) 27 种特征按照整数编码的大小从前往后排列

  得到的 one-hot 编码如下:



	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	空
h	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
l	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
o	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
空	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
w	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
o	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
r	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
l	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
d	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

- 例 2：对 ["中国", "美国", "日本", "美国"] 进行 one-hot 编码
  1. 确定要**编码的对象--["中国", "美国", "日本", "美国"]**
  2. 确定**分类变量 — 中国，美国，日本，**共 3 种类别
  3. 以上问题就相当于，**有 3 个样本，每个样本有 3 个特征，**将其转化为**二进制向量**表示
  4. 我们首先进行**特征的整数编码**：中国-0，美国-1，日本-2，并将特征按照从小到大排列

  得到 one-hot 编码如下：

  ["中国", "美国", "日本", "美国"] -> [[1,0,0], [0,1,0], [0,0,1], [0,1,0]]

-为什么要 one-hot？

- one hot 编码是将类别变量转换为机器学习算法易于利用的一种形式的过程。
- 上面的 hello world 相当于多分类的问题（27 分类），每个样本只对应于一个类别（即只在对应的特征处值为 1，其余地方值为 0），而我们的分类结果，得到的往往是隶属于某个类别的概率，这样在进行损失函数（例如交叉熵损失）或准确率计算时，变得非常方便

-one-hot 编码的缺陷

- one-hot 编码要求每个类别之间相互独立，如果之间存在某种连续型的关系，或许使用 distributed representation（分布式）更加合适。