招聘信息文本分类 (MLP)

用MLP神经网络对招聘数据进行分类，从而训练出一个可以分类招聘信息的神经网络模型。

	PositionType	Job_Description	label
0	项目管理	\r\n 岗位职责： \r\n 1 、熟练使用 axure,visio ， 熟悉竞品分析， ...	0
1	项目管理	\r\n 岗位职责： \r\n 1 、熟练使用 axure,visio ， 熟悉竞品分析， ...	0
2	移动开发	\r\n 岗位职责： \r\n 1.负责安卓客户端应用的框架设计； \r\n 2.负责安卓客...	1
3	移动开发	\r\n 现诚招资深iOS高级软件开发工程师一枚！ 【你的工作职责】 1、负责iPhone手...	1
4	后端开发	\r\n 岗位职责： \r\n 1、基于海量交通信息数据的数据仓库建设、数据应用开发。 2、...	2

- 给了岗位描述，如何将它们分成不同的工作岗位
- 怎么将句子分成词语？词语怎么表示成向量？

$y = f(x)$ → $x \in R^n, y \in N$ 分类

$y = f(x)$ → $x \in R^n, y \in R$ 回归

计算机是无法直接处理文本信息的，所以，在我们构建神经网络之前，要对文本进行一定的处理。

相信大家对独热编码（one-hot encode）应该不陌生了，虽说它能把所有文本用数字表示出来，但是表示文本的矩阵会非常的稀疏，极大得浪费了空间，而且这样一个矩阵放入神经网络训练也会耗费相当多的时间。

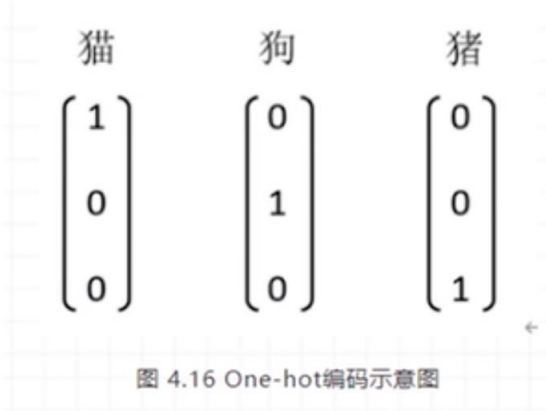
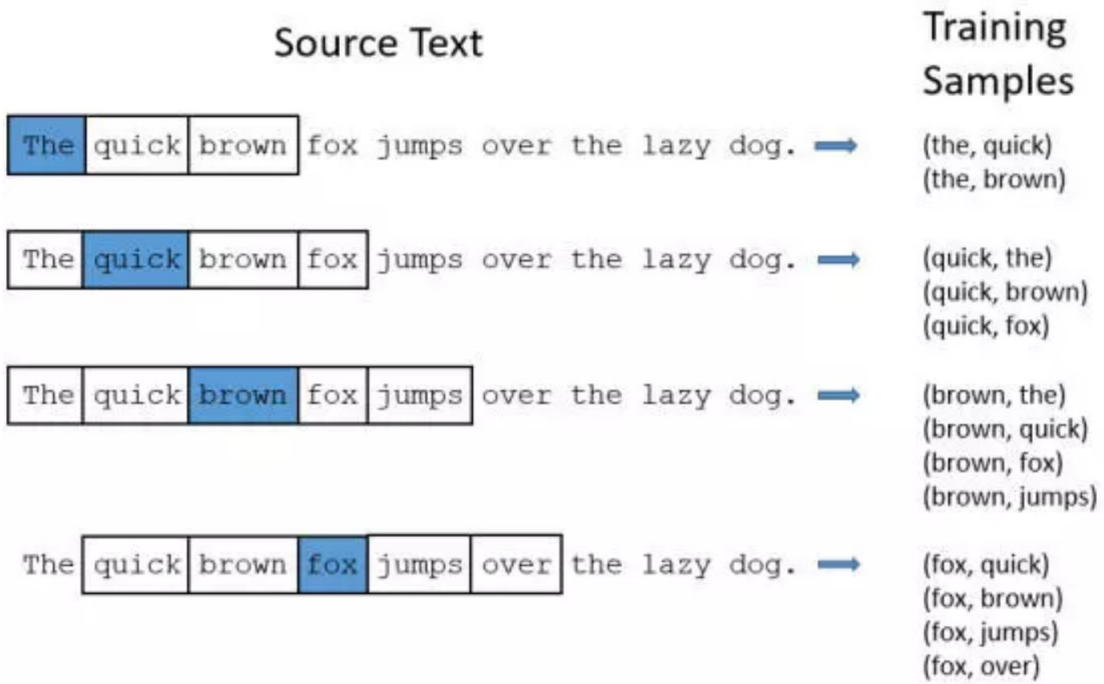


图 4.16 One-hot编码示意图

- 【如何使用one-hot】
  - 假设词典中不同词的数量（词典大小）为N，每个词可以和从0到$N-1$的连续整数一一对应。这些与词对应的整数叫作词的索引。
  - 假设一个词的索引为i，为了得到该词的one-hot向量表示，我们创建一个全0的长为N的向量，并将其第i位设成1。这样一来，每个词就表示成了一个长度为N的向量，可以直接被神经网络使用。
  - **简单来说：就是有多少个不同的词，我就会创建多少维的向量，如上：一个词典中有N个不同词，那么就会开创N维的向量，其中单词出现的位置为以1，该位置设为i，那么对应的向量就生成了。举个例子：[我，喜，欢，学，习]，其中的“我”就可以编码为：[1,0,0,0,0]，后面的“喜”就可以编码为：[0,1,0,0,0]，依次类推。**
- 【存在的问题】
  - 无法使用该方法进行单词之间的相似度计算。
  - 原因就是每个单词在空间中都是正交的向量，彼此之间没有任何联系。
  - 比如我们通过余弦相似度进行度量。

$$\frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} \in [-1, 1].$$

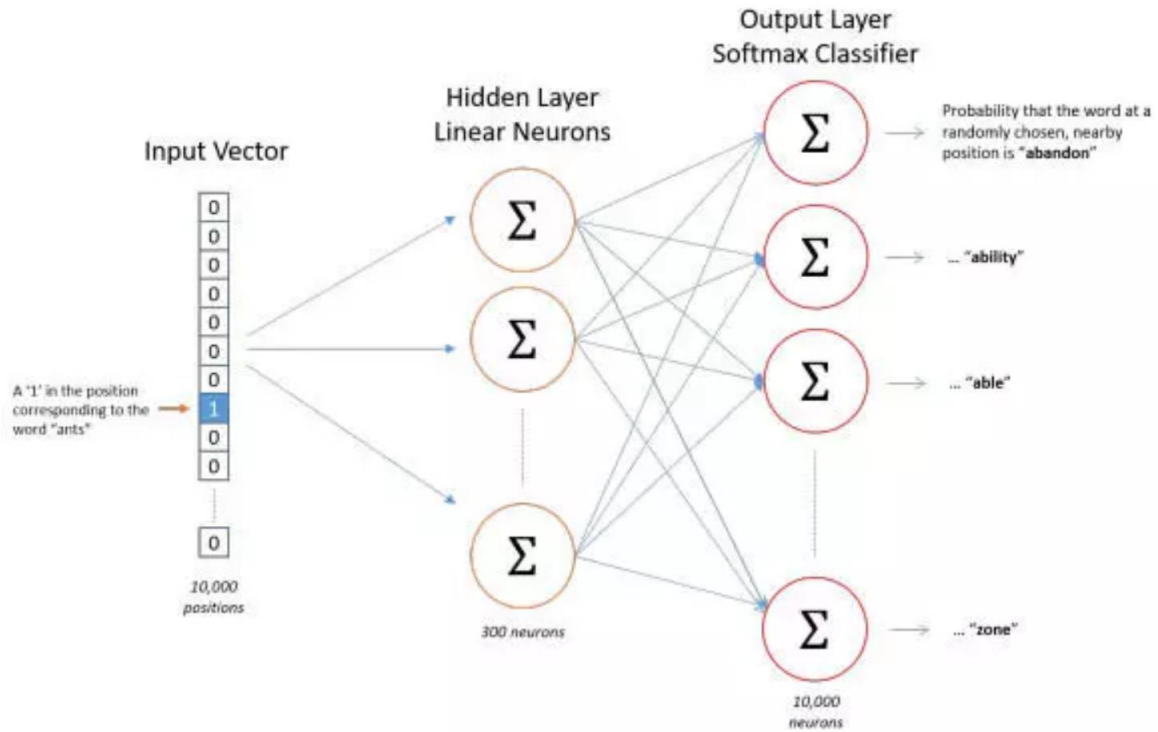
  - 对于向量$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$，它们的余弦相似度是它们之间夹角的余弦值。
- 【解决策略】
  - 既然one-hot的方式没有办法解决，那么我们就需要通过词嵌入的方式来解决。也就是我们后面重点要讲解的word2vec的方法。该方法目前有两种实现模型。
    - 跳字模型（skip-gram）：通过中心词来推断上下文一定窗口内的单词。
    - 连续词袋模型（continuous bag of words，CBOW）：通过上下文来推断中心词。



一文详解 Word2vec 之 Skip-Gram 模型（结构篇）

Skip-gram的数据准备

模型的输入如果为一个10000维的向量，那么输出也是一个10000维度（词汇表的大小）的向量，它包含了10000个概率，每一个概率代表着当前词是输入样本中output word的概率大小。
下图是神经网络的结构：



Skip-Gram 模型

隐层的节点数据就是我们的表征向量

2. 中文分词之jieba分词

中文文本处理会比处理英文多一步，中文词与词之间并不是用“空格”分开的，计算机不能处理这么高度抽象的文字，所以我们得通过一个Python库比如jieba来将中文文本进行分词，然后用“空格”将词分开，形成类似英文那样的文本，方便计算机处理。jieba分词示例图如图 4.59所示。

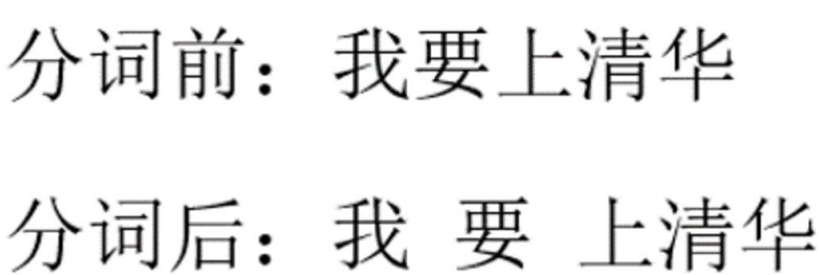


图 4.59 jieba分词示例图

```
1. # 打开操作系统的命令行，输入安装指令
2. pip install jieba
```

```
# ----------------------  代码布局：  ----------------------
# 1、导入 Keras, matplotlib, numpy, sklearn 和 panda的包
# 2、招聘数据数据导入
# 3、分词和提取关键词
# 4、建立字典，并使用
# 5、训练模型
# 6、保存模型，显示运行结果
# ----------------------  代码布局：  ----------------------
```