

# 如何用 word embedding 计算你和女神的距离？

原创 宫业奇 LeetCode力扣 3月19日

点击上方蓝字关注我们 ✨

下面开始今天的学习 ~



在 NLP (Natural Language Processing, 自然语言处理) 领域, word embedding 已经成为了众所周知的技术。如果你想要从事数据/算法相关工作, 它几乎成为了绕不开的坎。在面试环节, 如果你的岗位与 NLP 相关, 那 word embedding 就是必考内容, 请认真做笔记; 面试其他机器学习领域掌握 word embedding 也是强有力的加分项。

在现实生活中 word embedding 已经拥有非常广泛的应用: 语音助手、机器翻译、情感分析... 因为 word embedding 的特殊性, 它几乎覆盖了所有 NLP 的应用。

对于其他开发者, 尝试 word embedding 也有无穷的乐趣, 想知道你和女神的距离吗? 求一下两个词的余弦相似度即可。

其实相比于较为复杂的深度学习模型, word embedding 相对简单得多, 但是鲜有中文文章能将其讲解透彻。本文将会从传统的 one-hot 编码开始, 阐述其优劣, 并延伸至 word embedding 技术和其优点, 到这里, 所有的读者都可以容易的理解。对于想深入了解算法本身的读者, 下篇会继续阐述一些算法的底层细节, 包括算法的 forward prop 和 backward prop, 以及其意义。



**为什么要让单词变成向量**

**让计算机知道爱**

人类可以很轻易地理解一个单词、词组或者字母，比如「LOVE」，但机器是理解不了的。想要让机器理解单词，就必须要把它变成一串数字（向量）。下面介绍的 One-Hot Encoding（One-Hot 编码）和 Word Embedding（词嵌入）和就是把单词变成向量的两类方法。



## One-Hot Encoding

### 不负责任的老师

以前人们采用的方法是 one-hot encoding。

我们以英文为例，首先你要维护一个很长很长很长的词汇表，词汇表可以是前人总结出来的常用词，也可以是你文本数据里的所有单词的集合，词汇表大概长成这样：

$$\begin{bmatrix} a \\ \cdot \\ \cdot \\ love \\ \cdot \\ \cdot \\ \cdot \\ zulu \end{bmatrix}$$

可以把词汇表理解成一个  $(V, 1)$  维的向量，其中  $V$  为词汇个数，在上面这个词汇表里，第一个词是  $a$ ，第  $v$  个词是  $zulu$ ，假设  $love$  是其中的第 520 个词，那么  $love$  这个单词就可以表示成如下向量：

$$\begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

即在这个  $(V, 1)$  维的向量中，第 520 元素为 1（表示出现了单词 love），其余元素为 0。同理，词汇表中的任何一个单词都可以以这种形式表达，这个方法叫做 one-hot encoding。



## word embedding

### 让计算机学会爱

one-hot encoding 可以让计算机知道有这么个单词，但这个单词表示什么意思？和其他单词有什么关联？计算机是理解不了的。比如「love」和「romantic」，人类可以很轻易的理解这两个单词，但是 one-hot encoding 的结果只能告诉计算机：「love」和「romantic」仅仅是非常高维的空间里两个毫无关系的向量（内积为 0）。

one-hot encoding 不是一个好老师，它只让计算机死记硬背单词，却不能让计算机理解单词背后的文化和内涵。

此时就需要 word embedding 这个优秀的老师登场了。

word embedding 将 one-hot encoding 的向量映射到一个新的空间，在这个空间里，「love」和「romantic」、「apple」和「orange」等不再是毫无关系的高维向量，表示近似含义的单词可能会更加接近彼此，向量间的相似度也更有意义。

比如说，经过了 word embedding 后，每个单词都会映射到一个 300 维的空间，那么单词可能会被表示成如下形式：

$$\begin{matrix} King = \begin{bmatrix} -0.95 \\ 0.93 \\ 0.7 \\ \vdots \\ \vdots \end{bmatrix} & Queen = \begin{bmatrix} 0.97 \\ 0.95 \\ 0.69 \\ \vdots \\ \vdots \end{bmatrix} & Apple = \begin{bmatrix} 0.00 \\ -0.01 \\ 0.03 \\ \vdots \\ \vdots \end{bmatrix} & Orange = \begin{bmatrix} 0.01 \\ 0.00 \\ -0.02 \\ \vdots \\ \vdots \end{bmatrix} & Man = \begin{bmatrix} -1 \\ 0.01 \\ 0.03 \\ \vdots \\ \vdots \end{bmatrix} & Woman = \begin{bmatrix} 1 \\ 0.02 \\ 0.02 \\ \vdots \\ \vdots \end{bmatrix} \end{matrix}$$

第一个维度表示性别，King 接近 -1（男性），Queen 接近 +1（女性），而 Apple 和 Orange 与性别没什么关系，所以接近 0；第二个维度表示尊贵程度，第三个维度表示年龄，可以以此类推。

显而易见，经过 word embedding 后，「King」和「Queen」更接近了，「Apple」和「Orange」更接近了，同时「King」或「Queen」离「Apple」或「Orange」更远了。

现在计算机已经懂了，后来学者们又发现，其实通过向量的基础运算，我们也可以对 word embedding 的结果更懂一些，比如说，在上面的例子中，代表 King 的向量减去代表 Queen 的向量，其结果和代表 Man 的向量减去代表 Woman 的向量近似相等：

$$\mathbf{x}_{King} - \mathbf{x}_{Queen} \approx \mathbf{x}_{Man} - \mathbf{x}_{Woman}$$

再比如假如以 B 站的网页信息和弹幕等作为语料库训练模型，二次元的女神也可以被作为词向量进行计算：

$$\mathbf{x}_{雷姆} - \mathbf{x}_{程序员} \approx \mathbf{x}_{二次元} - \mathbf{x}_{现实}$$

这是一个很惊奇的发现：原来单词的基础运算也有一些奇妙的意义。现在我们不仅懂了，还很扎心了。

最后还需要澄清一点，在本文举的例子中，word embedding 的结果，每个维度都有很容易解释的意义，比如性别等，实际上算法计算出的词向量其 **维度代表的意义往往难以解释，也不具备现实意义**。这也是深度学习一直很魔幻的地方：算法研究者自己计算出的结果自己都很难解读。



## 总结

最后我们来做一个总结，自然语言的向量化表示方法主要有两类：one-hot encoding 和 word embedding。它们的优缺点如下：

	One-Hot Encoding	Word Embedding
优点	简单；	可以很好的表达单词之间的联系；维度低，好计算；方便迁移到不同 NLP 任务中
缺点	无法表达词语之间的联系；表达过于稀疏	对每个维度的意义难以解释

本文对 word embedding 的意义和结果做了阐述，关于 word embedding 的具体实现我们下篇文章见。

## 参考资料

Sequence Model - deeplearning.ai

## BY /

本文作者：宫业奇

编辑&版式：霍霍

声明：本文归“力扣”版权所有，如需转载请联系。文章封面图来源于网络，如有侵权联系删除。



### 推荐阅读

#### 力扣新题库《程序员面试金典（第 6 版）》

开始练习

- Cracking the Code Interview



#### 力扣新题库《剑指 Offer（第二版）》

开始练习



#### 力扣精选 2020 名企高频面试题

开始练习



#### 「正则表达式」王国奇遇记

点击查看

