情感分类 (2)：词嵌入

1 one-hot 编码

在多分类任务中，常用 one-hot 编码，中文可以翻译为“独热”。假设词表中有“他、是、男、生”四个字，one-hot 编码就是给这四个字用 0-1 编码：

他	(1, 0, 0, 0)
是	(0, 1, 0, 0)
男	(0, 0, 1, 0)
生	(0, 0, 0, 1)

如果词表很大的话，那么向量的维数便很大，而且任意向量都是正交的，无法表示两个字的相似性。既然如此，是否可以直接用一个整数表示呢？也不行，虽然可以减少向量的维数，但是会存在如下问题：

- 神经网络的输入是向量。
- 也很难表示两个字的相似性。
- 一个整数没有任何几何意义。

2 词嵌入

为了解决上述问题，便有了词嵌入，即使用密集的实数向量。one-hot 编码得到的向量是二进制的、稀疏的 (绝大部分元素都是 0)、维度很高的 (维度大小等于词表中的单词个数)，而词嵌入是低维的是数向量 (即密集向量，与稀疏向量相对)，参见图1。与 one-hot 编码得到的词向量不同，词嵌入是从数据中学习得到的。常见的词向量维度是 256、512 或 1024 (处理非常大的词表时)。与此相对，one-hot 编码的词向量维度通常为 20,000 或更高 (对应包含 20,000 个标记的词表)。因此，词向量可以将更多的信息塞入更低的维度中。而且词嵌入很好地解决 one-hot 编码的两个问题。

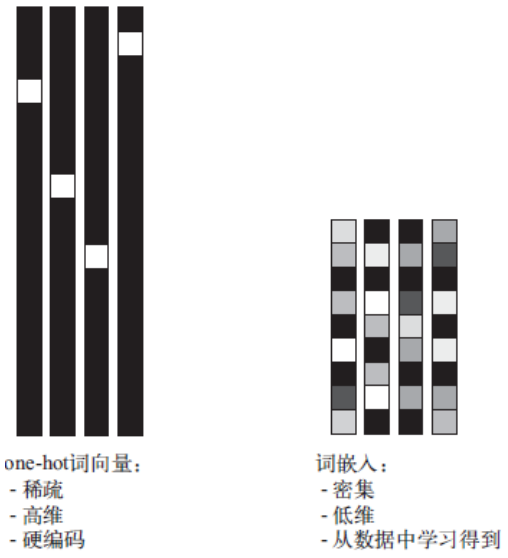

图 1 one-hot 编码或 one-hot 散列得到的词表示是稀疏的、高维的、硬编码的，而词嵌入是密集的、相对低维的，而且是从数据中学习得到的。

3 Keras 中的 Embedding 层

```
tf.keras.layers.Embedding(
    input_dim, output_dim, embeddings_initializer='uniform',
    embeddings_regularizer=None, activity_regularizer=None,
    embeddings_constraint=None, mask_zero=False, input_length=None, **kwargs
)
```

图 2 Keras 中的 Embedding 层

输入张量的形状是 (b, input_length)，输出是 (b, input_length, hidden_dim)。其中参数“input_dim”指词表大小 (即包含多少个字)，参数“output_dim”指实数向量的维数，与 hidden_dim 对应，参数“input_length”指输入字序列的长度。