1. 词汇表征

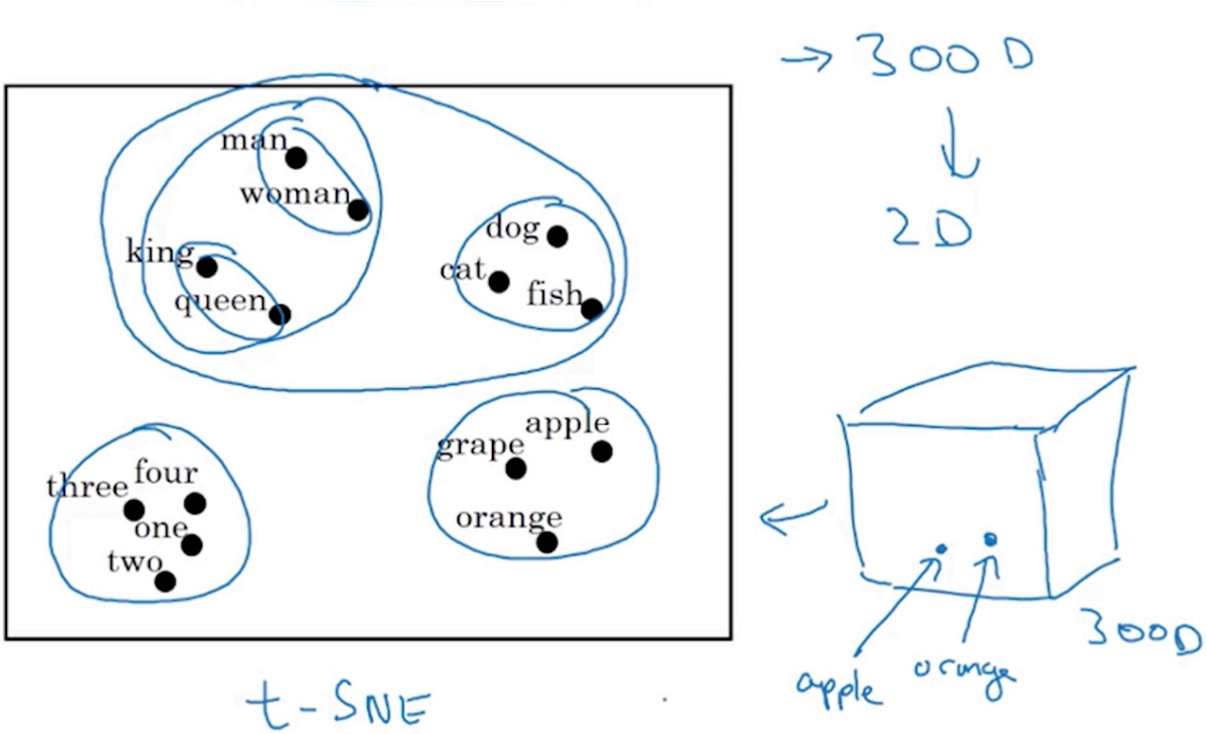
- **one-hot 表征：**

one-hot 向量将每个单词表示为完全独立的个体，不同词向量都是**正交**的，因此单词间的相似度无法体现。

- **词嵌入(word embedding)表征：**

是 NLP 中语言模型与表征学习技术的统称，概念上而言，它是指把一个维数为所有词的数量的高维空间（one-hot 形式表示的词）“嵌入”到一个维数低得多的连续向量空间中，每个单词或词组被映射为实数域上的向量。对大量词汇进行词嵌入后获得的词向量，可用于完成**命名实体识别（Named Entity Recognition）**等任务。

下面是用t-SNE算法将高维的词向量映射到2维平面上，进而对词向量进行可视化，可以看到相似的词总是聚在一块。



2. 使用Word Embedding

Word Embedding对不同单词进行了特征化的表示，因此用词向量做迁移学习可以降低学习成本，提高效率，其步骤如下：

1. 从大量的文本集中学习词嵌入，或者下载网上开源的、预训练好的词嵌入模型；
2. 将这些词嵌入模型迁移到新的、只有少量标注训练集的任务中；

3. 可以选择是否微调词嵌入。当标记数据集不是很大时可以省下这一步。

3. 词嵌入的特性

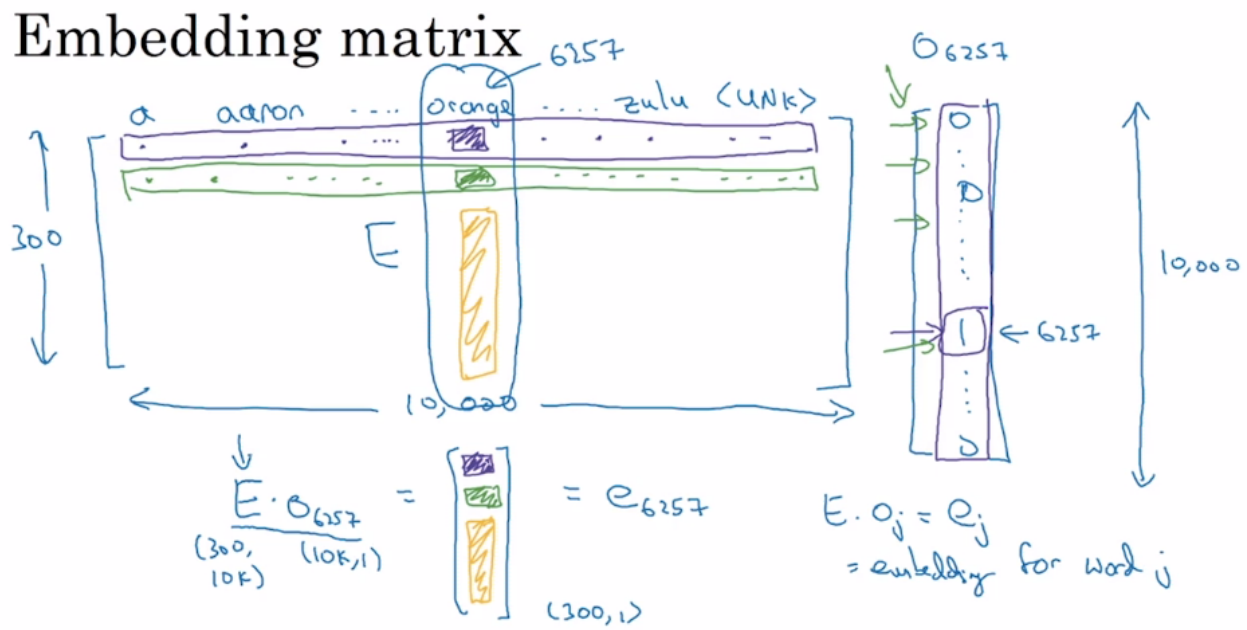
词嵌入的一个重要特性便是能够帮助实现类比推理。比如：$e_{man} - e_{woman} \approx e_{king} - e_{queen}$

余弦相似度：

$$sim(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$$

4. 嵌入矩阵

要对一个词汇表学习词嵌入，就是要学习这个词汇表对应的嵌入矩阵E。



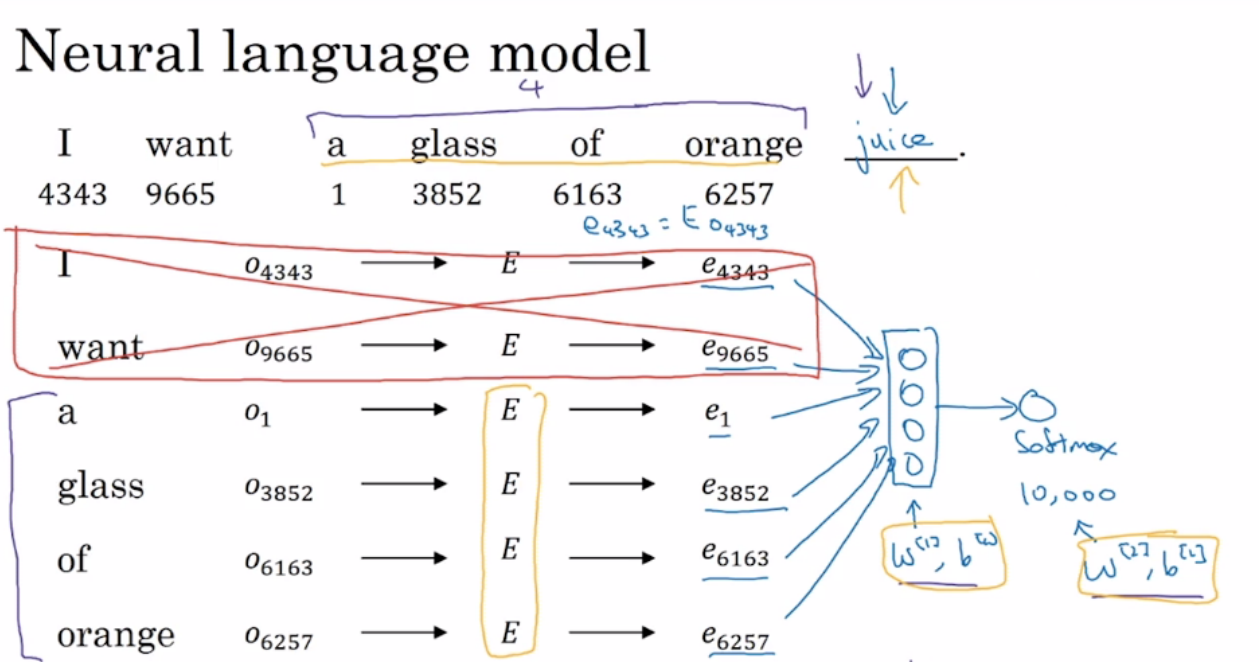
设字典中的第 i 个词的one-hot向量为 o_i，词嵌入为e_i，则有：

$$E \cdot o_i = e_i$$

但在实际情况下一般不这么做。因为 one-hot 向量维度很高，且几乎所有元素都是 0，这样做的效率太低。因此，实践中直接用专门的函数查找矩阵E 的特定列，比如Keras 中可以用 Embedding layer方便提取需要的列。

5. 学习词嵌入

神经概率语言模型（Neural Probabilistic Language Model）构建了一个通过上下文词预测目标词的神经网络，在训练这个语言模型的同时，得到词嵌入。



相关论文：[Bengio et. al., 2003, A neural probabilistic language model]

6. Word2Vec

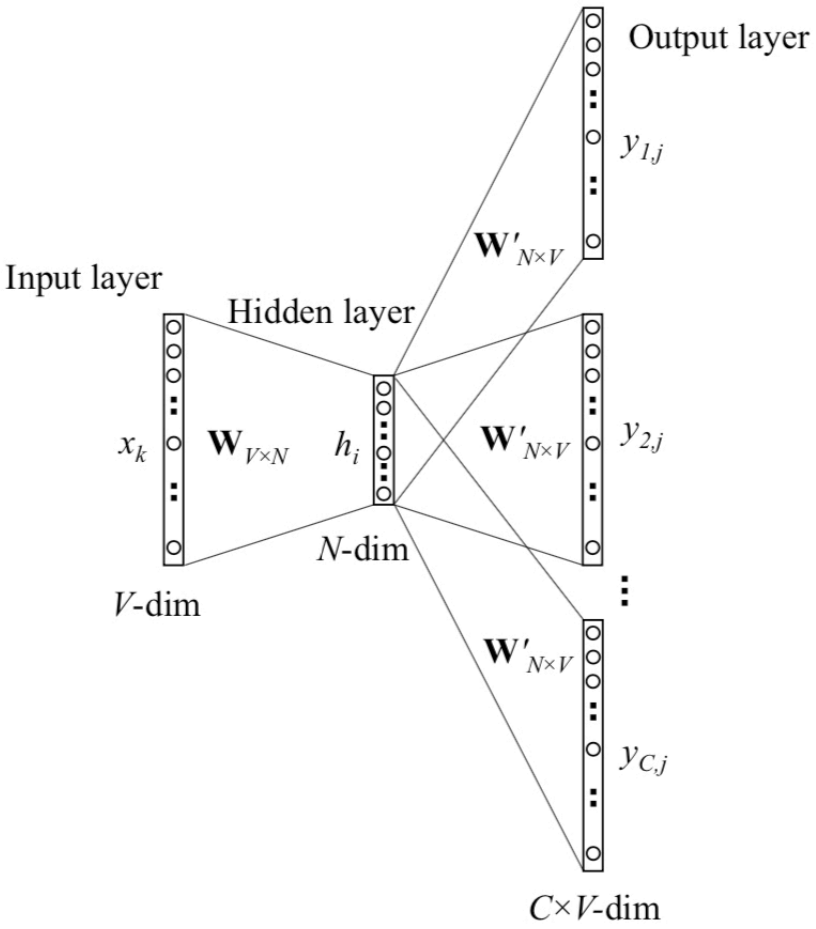
Word2Vec是一种简单高效的词嵌入学习算法，包括 2 种模型：

- **Skip-Gram:** 根据目标词预测上下文词；
- **CBOW:** Continuous Bag of Word, 根据上下文词预测目标词

而每种模型又包含**负采样(Negative Sampling)**和**层序softmax（Hierarchical Softmax）**两种训练方法。

训练神经网络时，隐藏层参数既是学到的词嵌入！

6.1 Skip-Gram



左边的input layer是目标词的one-hot向量，与词向量矩阵运算得到目标词的词嵌入：

$$e_c = E \cdot o_c$$

经过Softmax层得到目标上下文词的条件概率：

$$p(t|c) = \frac{\exp(\theta_t^T e_c)}{\sum_j^m \exp(\theta_j^T e_c)} \tag{1}$$

θ_t是一个与输出t 有关的参数，其中省去了偏置项。损失函数用交叉熵：

$$\mathcal{L}(y, \hat{y}) = -\sum_{i=1}^{m} y_i \log \hat{y}_i \tag{2}$$

6.2 Hierarchical Softmax

在Softmax层，由式（1）可以看到，每次计算条件概率，都需对字典中的所有词做求和计算，计算量太大！

简化方案是使用Hierarchical Softmax分类器，它相当于一个树形分类器，树的每个节点都是一个二分类器。一般使用Huffman树，即高频词在顶部，低频词在底部。

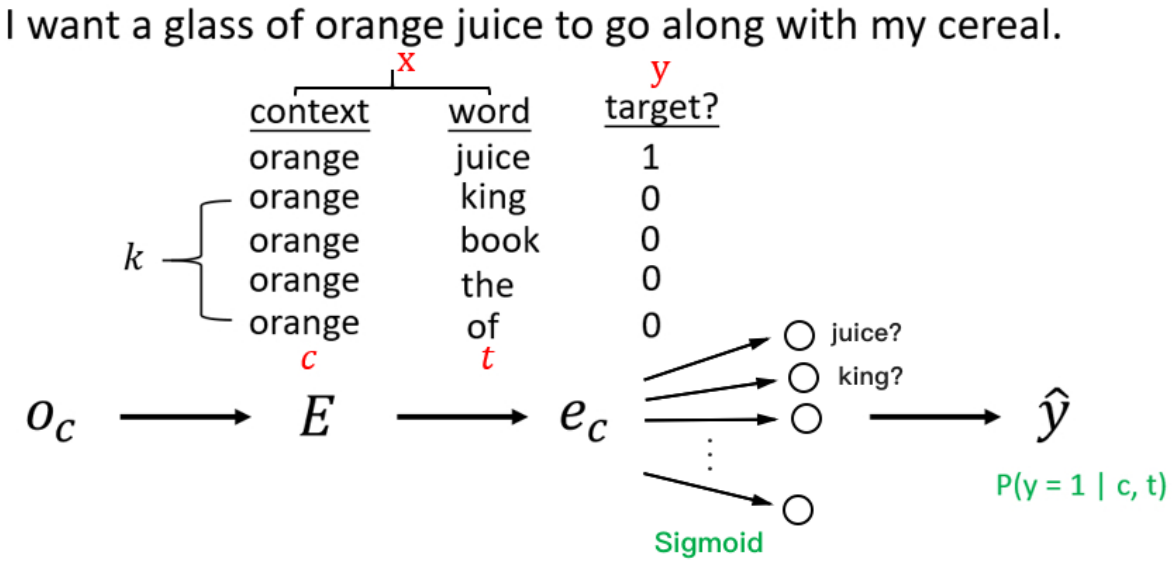
6.3 About Sampling

在构建上下文目标词对时，如何采样对模型有不同的影响：

- 对语料库均匀且随机采样：使得如the、of、a等这样的一些词会出现的相当频繁，导致上下文和目标词对经常出现这类词汇，但我们想要的目标词却很少出现。
- 采用不同的启发来平衡常见和不常见的词进行采样。word2vec作者采用的是**带权采样的方法**。

7. 负采样

所谓负采样，就是从所有上下文目标词对中，采样部分负样本来训练模型。



首先设正样例标签为1，采样k 个负样本，设标签为0.（对于小数据集，k可以取$5 \sim 20$；对于大数据集，k 可以取$2 \sim 5$）；

输出层改用sigmoid函数计算正样本的概率：

$$p(t|c) = \sigma(\theta_t^T e_c)$$

其中，θ_t、e_c分别是目标词和上下文词的词向量。

相比之前的softmax多分类需计算整个词表，现在只需计算$k+1$ 个sigmoid单元，计算量大大降低了。

选择某个词作为负样本的概率，作者采用如下经验公式：

$$p(w_i) = \frac{f(w_i)^{0.75}}{\sum_j^V f(w_j)^{0.75}}$$

其中，$f(w_i)$代表词w_i 在语料库中出现了频率。

8. GloVe

Glove模型基于语料库统计了词的共现矩阵X，元素X_{ij}表示词条j出现在词条i上下文的次数。基于词向量和共现矩阵，构建损失函数：

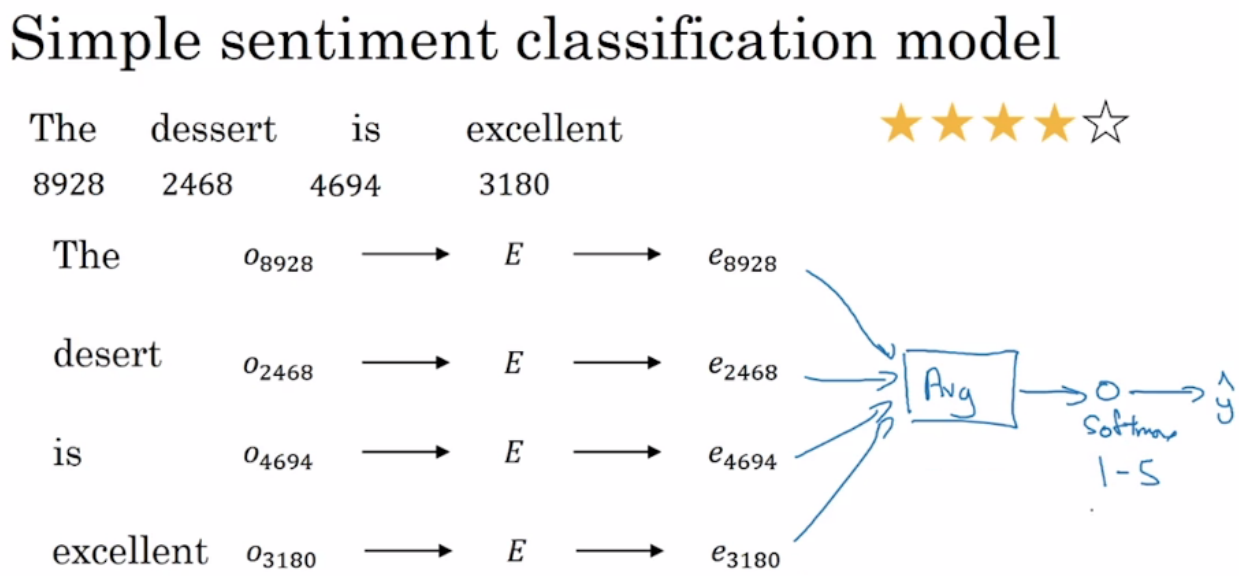
$$J=\sum_{i,j}^{N} f(X_{ij})\left(\theta_i^T e_j+b_i-b_j'-\log X_{ij}\right)^2$$

其中，θ_i、e_j是词i和词j的词向量。

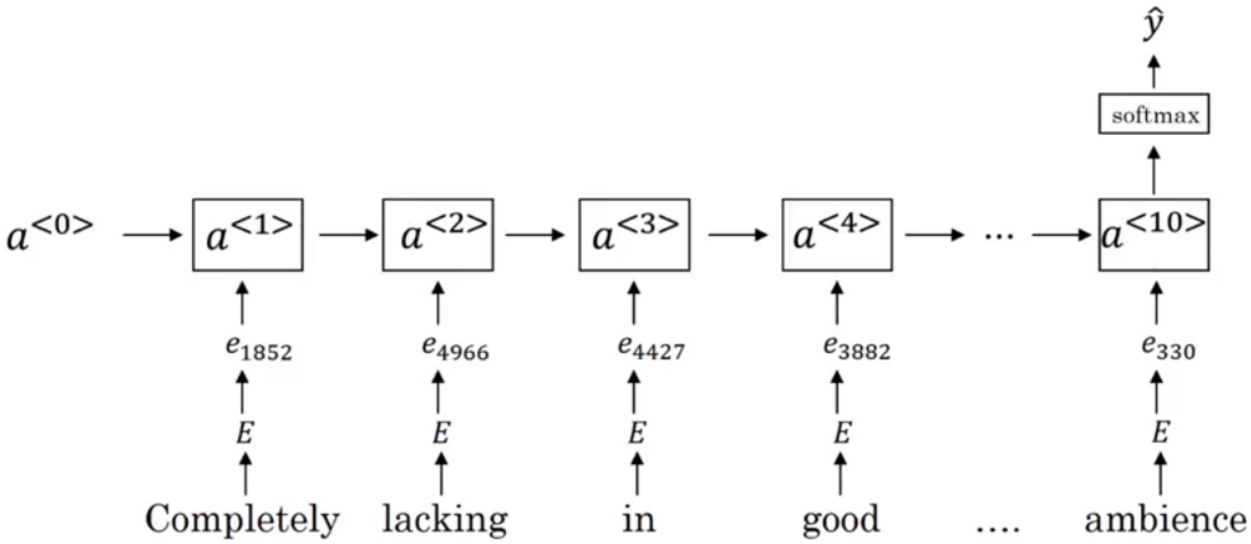
更多关于GloVe的内容见GloVe理解.

9. 情感分类

情感分类是指分析一段文本对某个对象的情感是正面的还是负面的，实际应用包括舆情分析、民意调查、产品意见调查等等。情感分类的问题之一是标记好的训练数据不足。但是有了词嵌入得到的词向量，中等规模的标记训练数据也能构建出一个效果不错的情感分类器。



如上图所示，用词嵌入方法获得嵌入矩阵 EE 后，计算出句中每个单词的词向量并取平均值，输入一个 Softmax 单元，输出预测结果。这种方法的优点是适用于任何长度的文本；缺点是没有考虑词的顺序，对于包含了多个正面评价词的负面评价，很容易预测到错误结果。


$\hat{y}$
softmax
$a^{<0>}$
$a^{<1>}$
$a^{<2>}$
$a^{<3>}$
$a^{<4>}$
...
$a^{<10>}$
e_{1852}
e_{4966}
e_{4427}
e_{3882}
e_{330}
E
E
E
E
E
Completely
lacking
in
good
....
ambience