deeplearning.ai

吴恩达 DeepLearning.ai 课程提炼笔记（5-2）序列模型 --- NLP和词嵌入

以下为吴恩达老师 DeepLearning.ai 课程项目中，第五部分《序列模型》第二周课程“**NLP和词嵌入**”关键点的笔记。

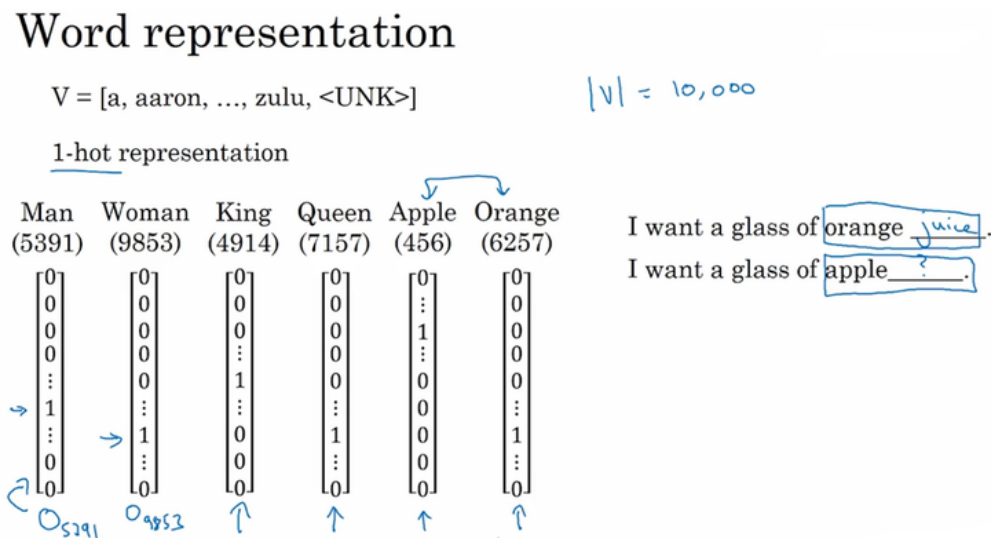
---

1. 词汇表征

在前面学习的内容中，我们表征词汇是直接使用英文单词来进行表征的，但是对于计算机来说，是无法直接认识单词的。为了让计算机能够能更好地理解我们的语言，建立更好的语言模型，我们需要将词汇进行表征。下面是几种不同的词汇表征方式：

one-hot 表征：

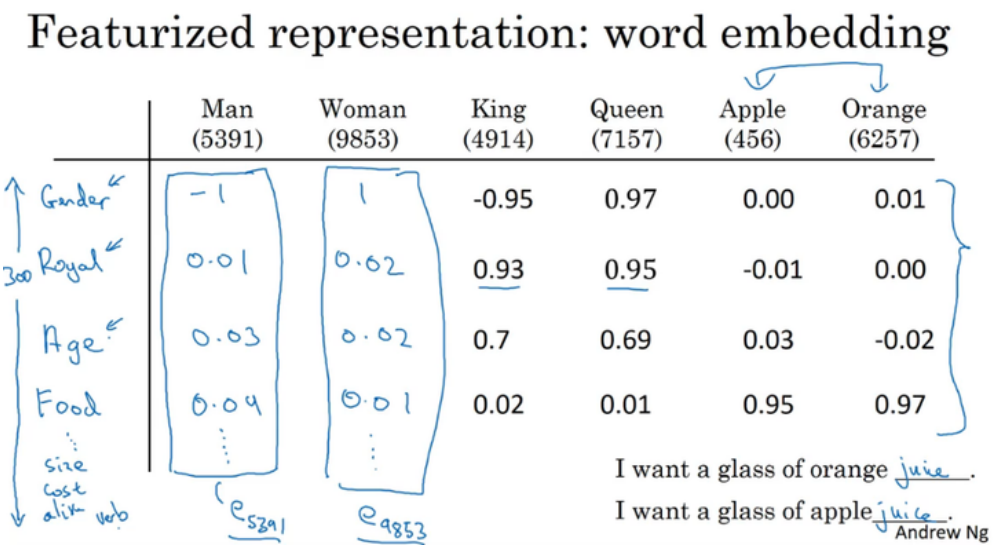
在前面的一节课程中，已经使用过了one-hot表征的方式对模型字典中的单词进行表征，对应单词的位置用1表示，其余位置用0表示，如下图所示：



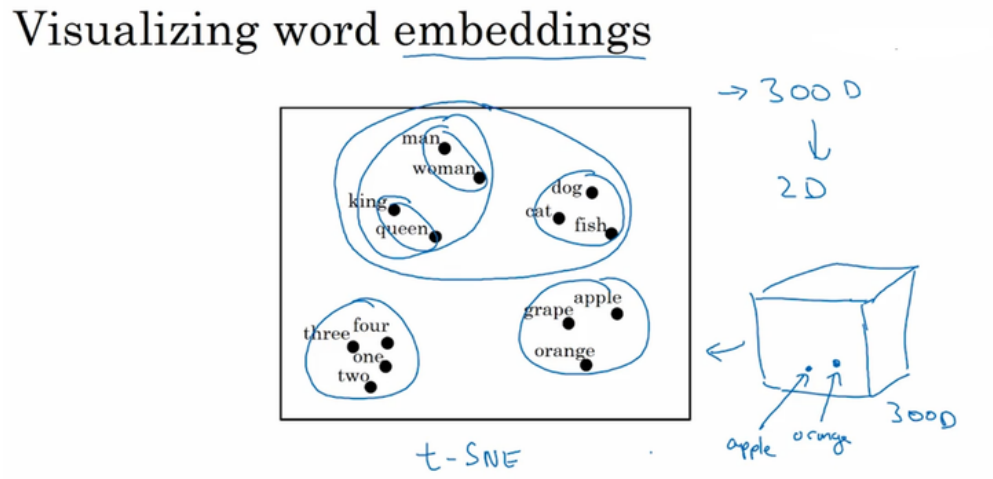
one-hot表征的缺点：这种方法将每个词孤立起来，使得模型对相关词的泛化能力不强。每个词向量之间的距离都一样，乘积均为0，所以无法获取词与词之间的相似性和关联性。

特征表征：词嵌入

用不同的特征来对各个词汇进行表征，相对与不同的特征，不同的单词均有不同的值。如下例所示：



这种表征方式使得词与词之间的相似性很容易地表征出来，这样对于不同的单词，模型的泛化性能会好很多。下面是使用t-SNE算法将高维的词向量映射到2维空间，进而对词向量进行可视化，很明显我们可以看出对于相似的词总是聚集在一块儿：

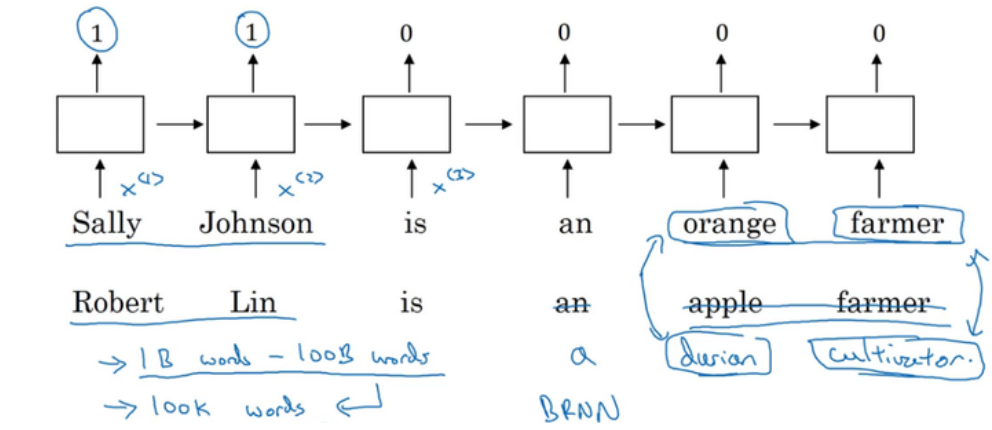


2. 使用 Word Embeddings

Word Embeddings对不同单词进行了实现了特征化的表示，那么如何将这种表示方法应用到自然语言处理的应用中呢？

名字实体识别的例子：

如下面的一个句子中名字实体的定位识别问题，假如我们有一个比较小的数据集，可能不包含durain（榴莲）和cultivator（培育家）这样的词汇，那么我们就很难从包含这两个词汇的句子中识别名字实体。但是如果我们从网上的其他地方获取了一个学习好的word Embedding，它将告诉我们榴莲是一种水果，并且培育家和农民相似，那么我们就有可能从我们少量的训练集中，归纳出没有见过的词汇中的名字实体。



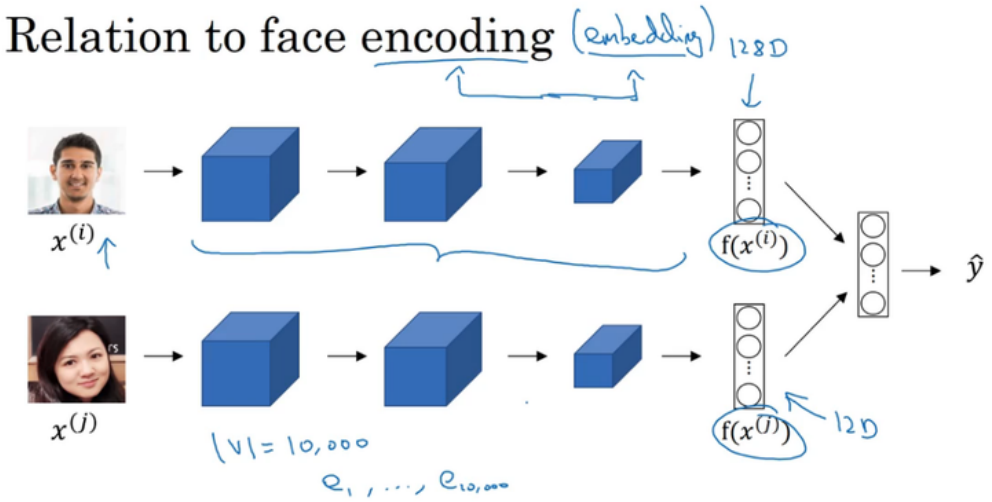
词嵌入的迁移学习：

有了词嵌入，就可以让我们能够使用迁移学习，通过网上大量的无标签的文本中学习到的知识，应用到我们少量文本训练集的任务中。下面是做词嵌入迁移学习的步骤：

- 第一步：从大量的文本集合中学习word Embeddings（1-100B words），或者从网上下载预训练好的词嵌入模型；
- 第二步：将词嵌入模型迁移到我们小训练集的新任务上；
- 第三步：可选，使用我们新的标记数据对词嵌入模型继续进行微调。

词嵌入和人脸编码：

词嵌入和人脸编码之间有很奇妙的联系。在人脸识别领域，我们会将人脸图片预编码成不同的编码向量，以表示不同的人脸，进而在识别的过程中使用编码来进行比对识别。词嵌入则和人脸编码有一定的相似性。

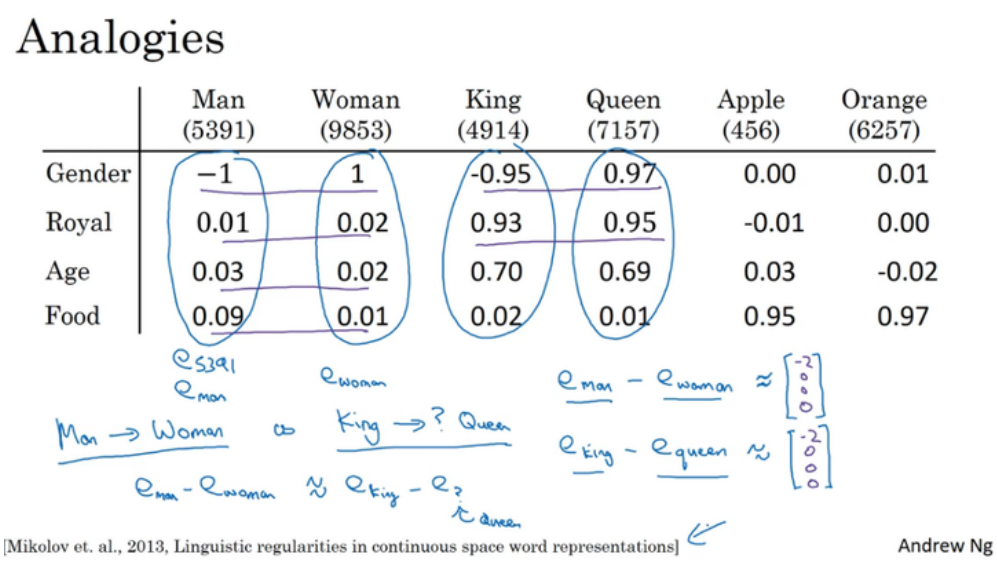


但是不同的是，对于人脸识别，我们可以将任意一个没有见过的人脸照片输入到我们构建的网络中，则可输出一个对应的人脸编码。而在词嵌入模型中，所有词汇的编码是在一个固定的词汇表中进行学习单词的编码以及其之间的关系的。

3. 词嵌入的特性

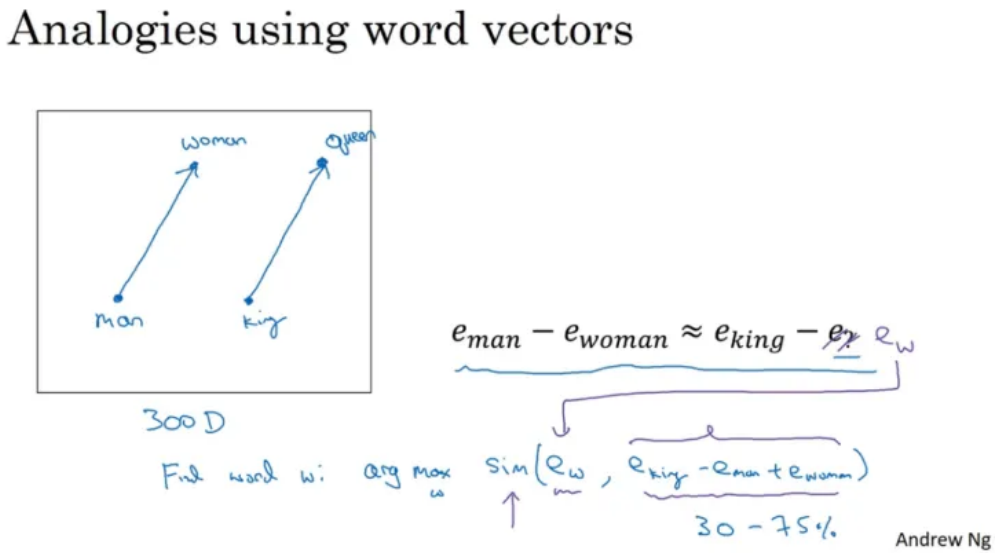
类比推理特性：

词嵌入还有一个重要的特性，它还能够帮助实现类比推理。如下面的例子中，通过不同词向量之间的相减计算，可以发现不同词之间的类比关系，man——woman、king——queen，如下图所示：



这种思想帮助研究者们对词嵌入建立了更加深刻的理解和认识。

计算词与词之间的相似度，实际上是在多维空间中，寻找词向量之间各个维度的距离相似度。



以上面的单词为例：

$$e_{man} - e_{woman} \approx e_{king} - e_{?}$$

对于上面的式子，我们寻找 $e_{?}$ ，*则相当于寻找下面两个结果的向量之间的最大相似度：*

$$arg\max_{?} sim(e_{?}, e_{king} - e_{man} + e_{woman})$$

相似度函数：

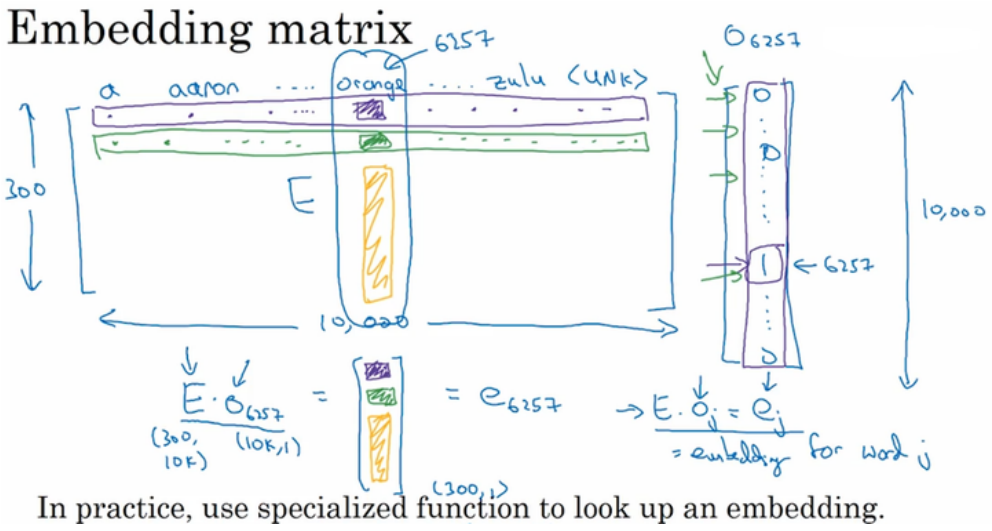
- 余弦相似度函数（Cosine similarity）：也就是向量 u 和 v 的内积

  $$sim(u,v) = \frac{u^T v}{||u||_2 ||v||_2}$$
- 欧氏距离：

  $$||u-v||^2$$

4. 嵌入矩阵

在我们要对一个词汇表学习词嵌入模型时，实质上就是要学习这个词汇表对应的一个嵌入矩阵 E 。当我们学习好了这样一个嵌入矩阵后，通过嵌入矩阵与对应词的one-hot向量相乘，则可得到该词汇的embedding，如下图所示：



In practice, use specialized function to look up an embedding.

5. 学习词嵌入

词嵌入的学习算法随着时间的进行逐渐变得越来越简单。

早期的学习算法：

如下面的例子中，我们要通过前面几个单词，预测最后一个单词：

- 通过将每个单词的one-hot向量与嵌入矩阵相乘，得到相应的Embedding；
- 利用窗口控制影响预测结果的单词数量，并将窗口内单词的Embedding堆叠起来输入到神经网络中；
- 最后通过softmax层输出整个词汇表各个单词可能的概率；
- 其中，隐藏层和softmax层都有自己的参数，假设词汇表的大小为 (10000) ，每个单词的Embedding大小是 (300) ，历史窗口大小为 4 ，那么输入的大小即为 $(300 * 4 = 1200)$ ，softmax输出大小为词汇表大小 (10000) ；
- 整个模型的参数就是嵌入矩阵 E ，以及隐藏层和softmax层的参数 $w^{[1]}, b^{[1]}, w^{[2]}, b^{[2]}$ ；
- 可以利用反向传播算法进行梯度下降，来最大化训练集似然函数，不断地从语料库中预测最后一个词的输出。

在不断地训练过程中，算法会发现要想最好地拟合训练集，就要使得一些特性相似的词汇具有相似的特征向量，从而就得到了最后的词嵌入矩阵 E 。

其他的上下文和目标词对：

我们将要预测的单词称为目标词，其是通过一些上下文推导预测出来的。对于不同的问题，上下文的大小和长度以及选择的方法有所不同。

- 选取目标词之前的几个词；
- 选取目标词前后的几个词；
- 选取目标词前的一个词；
- 选取目标词附近的一个词，（一种Skip-Gram模型的思想）。

6. Word2Vec

Word2Vec算法是一种简单的计算更加高效的方式来实现对词嵌入的学习。

Skip-grams：

在Skip-grams模型中，我们需要抽取上下文（Content）和目标词（Target）配对，来构造一个监督学习问题。

上下文不一定是要目标词前面或者后面离得最近的几个单词，而是随机选择一个词作为上下文，同时在上下文的一定距离范围内随机选择另外一个词作为目标词。构造这样一个监督学习问题的目的，并不是想要解决监督学习问题本身，而是想要使用这个问题来学习一个好的词嵌入模型。

模型流程：

- 使用一个具有大量词汇的词汇表，如Vocab size = 10000k；
- 构建基本的监督学习问题，也就是构建上下文（C）和目标词（T）的映射关系：C——T；
- o_c（one-hot）——E（词嵌入矩阵）——$e_c = E * o_c$（词嵌入）——$Softmax$ 层——$\hat{y}$；
- $Softmax$：$p(t|c) = \frac{e^{\Theta_t^T e_c}}{\sum_{j=1}^{10000} e^{\Theta_j^T e_c}}$，其中$\Theta_t$ 是与输出 t 有关的参数；
- 损失函数：$L(\hat{y}, y) = -\sum_{i=1}^{10000} y_i \log \hat{y}_i$，这是在目标词 y 表示为one-hot向量时，常用的softmax损失函数。
- 通过反向传播梯度下降的训练过程，可以得到模型的参数 E 和softmax的参数。

存在的问题：

- 计算速度的问题，如在上面的 $Softmax$ 单元中，我们需要对所有10000个整个词汇表的词做求和计算，计算量庞大。
- 简化方案：使用分级softmax分类器（相当于一个树型分类器，每个节点都是可能是一个二分类器），其计算复杂度是前面的 $\log|v|$ 级别。在构造分级softmax分类器时，一般常用的词会放在树的顶部位置，而不常用的词则会放在树的更深处，其并不是一个平衡的二叉树。

如何采样上下文：

在构建上下文目标词对时，如何选择上下文与模型有不同的影响。

- 对语料库均匀且随机地采样：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。这种方法是实际使用的方法。

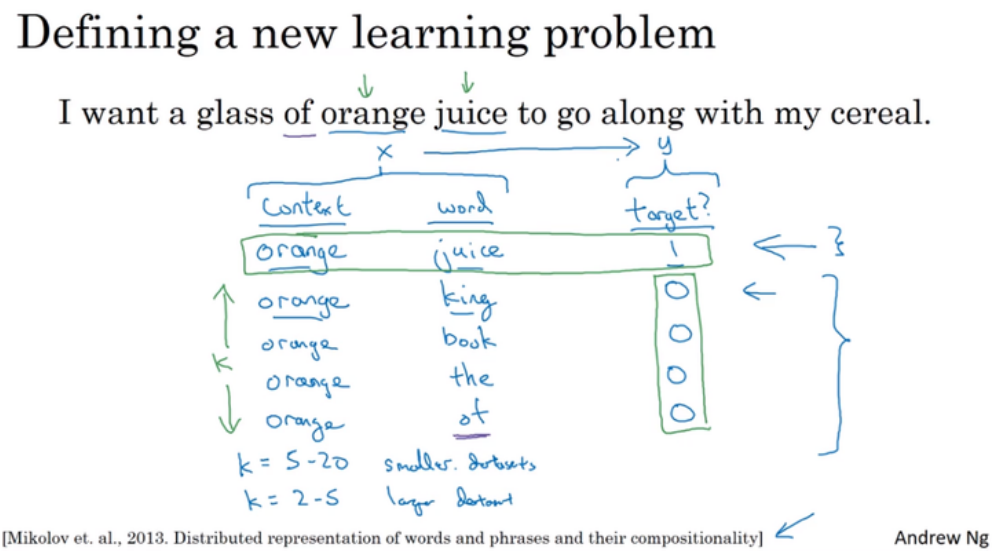
7. 负采样

Skip-grams模型可以帮助我们构造一个监督学习任务，将上下文映射到目标词上，从而让我们能够学习到一个实用的词嵌入模型。但是其缺点就是softmax计算的时间复杂度较高。下面介绍一种改善的学习问题：负采样。其能够做到的和Skip-grams模型相似，但其学习算法更加有效。

新的学习问题：

- 定义一个新的学习问题：预测两个词之间是否是上下文-目标词对，如果是词对，则学习的目标为1；否则为0。

- 使用k次相同的上下文，随机选择不同的目标词，并对相应的词对进行正负样本的标记，生成训练集。
- 建议：小数据集，k=5~20；大数据集，k=2~5。
- 最后学习 x—y 的映射关系。

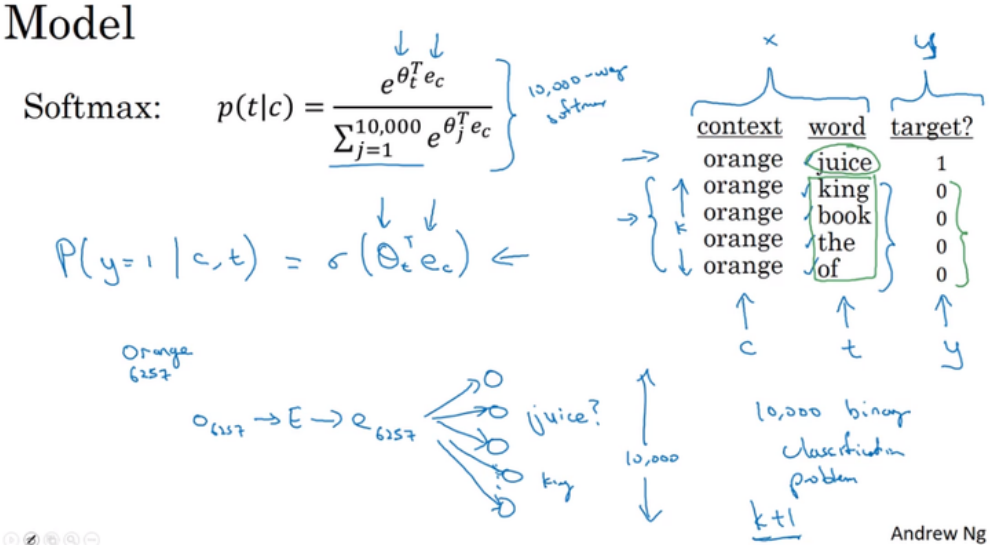


模型：

在负采样模型中，我们使用logistic回归模型：

$$P(y=1|c,t)=\sigma(\Theta_t^T e_c)$$

每个正样本均有k个对应的负样本。在训练的过程中，对于每个上下文词，我们就有对应的 $k+1$ 个分类器。如下图所示：



相比与Skip-grams模型，负采样不再使用一个具有词汇表大小时间复杂度高的庞大维度的Softmax，而是将其转换为词汇表大小个二分类问题。每个二分类任务都很容易解决，因为每个的训练样本均是1个正样本，外加k个负样本。

如何选择负样本：

在选定了上下文（Content）后，在确定正样本的情况下，我们还需要选择k个负样本以训练每个上下文的分类器。

- 通过单词出现的频率进行采样：导致一些类似a、the、of等词的频率较高；

- 均匀随机地抽取负样本：没有很好的代表性；
- **（推荐）**： $P(w_i)=\frac{f(w_i)^{3/4}}{\sum_{j=1}^{10000}f(w_j)^{3/4}}$ ，这种方法处于上面两种极端采样方法之间，即不用频率分布，也不用均匀分布，而采用的是对词频的 $3/4$ 除以词频 $3/4$ 整体的和进行采样的。其中， $f(w_j)$ 是语料库中观察到的某个词的词频。

8. GloVe 词向量

GloVe（global vectors for word representation）词向量模型是另外一种计算词嵌入的方法，虽然相比下没有Skip-grams模型用的多，但是相比这种模型却更加简单。

GloVe模型：

GloVe词向量模型中，要定义一个量 X_{ij} ，表示目标词 i 出现在上下文 j 的次数。模型的优化目标如下：

$$minimize \sum_{i=1}^{10000}\sum_{j=1}^{10000} f(X_{ij})(\Theta_i^T e_j+b_i+b_j-\log X_{ij})^2$$

- 其中，因为当 X_{ij} *为0时，* $\log X_{ij}$ 便没有意义，所以添加 $f(X_{ij})$ *的加权项，当* $X_{ij}=0$ 时，$f(X_{ij})=0$，*另外* $f(X_{ij})$ 对于一些频繁词和不频繁词有着启发式的平衡作用；
- 另外， $\Theta_i^T e_j$ 这一项中， Θ_i^T *和* e_j 都是需要学习的参数，在这个目标算法中二者是对称的关系，所以我们可以一致地初始化 Θ 和 e ，然后用梯度下降来最小化输出，在处理完所有词后，直接取二者的平均值作为词嵌入向量： $e_w^{final}=\frac{e_w+\Theta_w}{2}$ ，这与前面的算法有所不同。

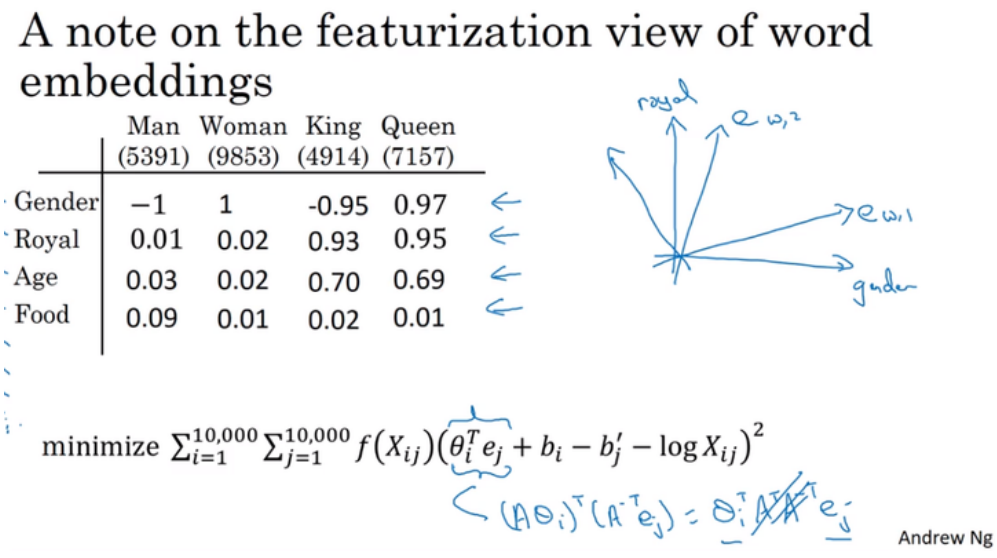
从上面的目标中，可以看出我们想要学习一些向量，他们的输出能够对上下文和目标两个词同时出现的频率进行很好的预测，从而得到我们想要的词嵌入向量。

词嵌入的特征化：

通过上面的很多算法得到的词嵌入向量，我们无法保证词嵌入向量的每个独立分量是能够让我们理解的。我们能够确定是每个分量是和我们所想的一些特征是有关联的，其可能是一些我们能够理解的特征的组合而构成的一个组合分量。使用上面的GloVe模型，从线性代数的角度解释如下：

$$\Theta_i^T e_j=\Theta_i^T A^T A^{-T} e_j=(A\Theta_i)^T(A^{-T}e_j)$$

加入的 A 项，可能构成任意的分量组合。



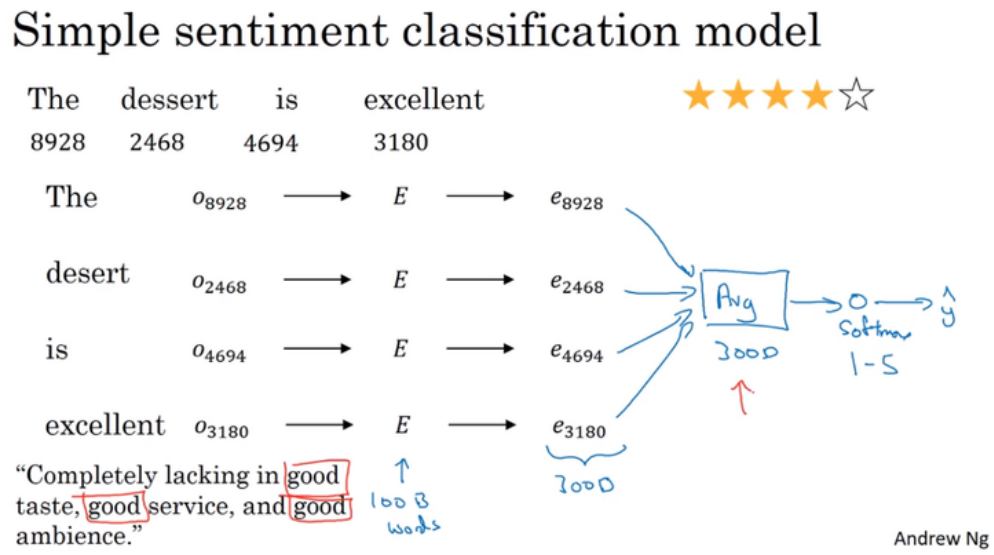
9. 情感分类

情感分类就是通过一段文本来判断这个文本中的内容是否喜欢其所讨论的内容，这是NLP中最重要的模块之一。

情感分类任务存在的一个问题就是只有很小的数据集，缺乏训练样本。但是在使用了词嵌入后，则能够带来很好的效果，足以训练一个良好的情感分类模型。

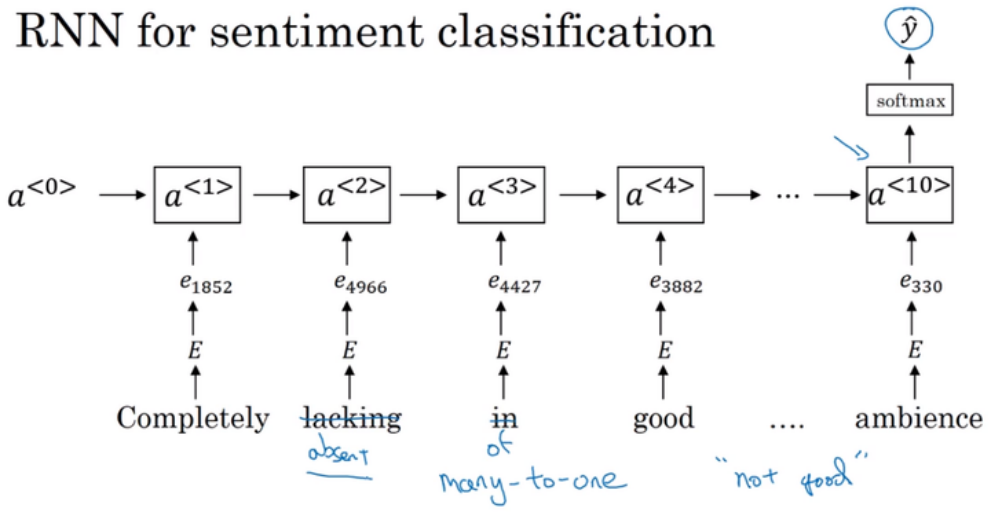
平均值或和的模型：

- 获取一个训练好的词嵌入矩阵 $\boldsymbol{E}$；
- 得到每个词的词嵌入向量，并对所有的词向量做平均或者求和；
- 输入到softmax分类器中，得到最后的输出 $\hat{\boldsymbol{y}}$；
- 缺点：没有考虑词序，可能会导致多数的积极词汇削弱前面消极词汇的影响，从而造成错误的预测。



RNN模型：

- 获取一个训练好的词嵌入矩阵 $\boldsymbol{E}$；
- 得到每个词的词嵌入向量，输入到many-to-one的RNN模型中；
- 通过最后的softmax分类器，得到最后的输出 $\hat{\boldsymbol{y}}$。
- 优点：考虑了词序，效果好很多。

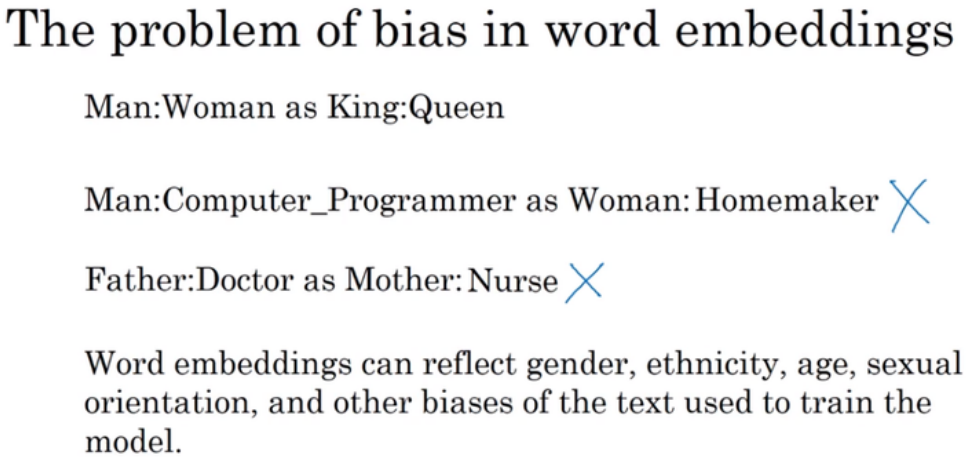


10. 词嵌入消除偏见

当下机器学习或者人工智能算法已经被应用到做一些非常重要的决策中，因此我们需要尽可能地保证其不受非预期形式的偏见的影响，如性别、种族歧视等等。下面介绍一些在词嵌入中消除偏见的办法。

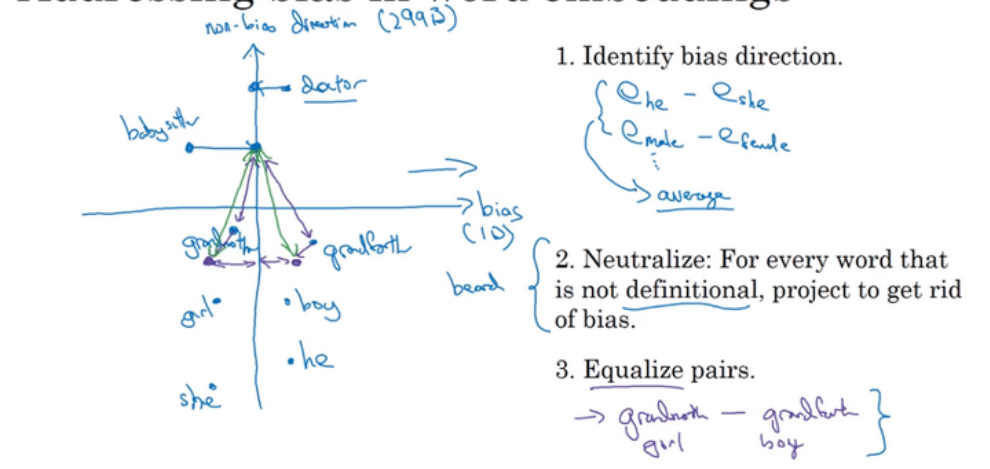
目前的偏见问题：

以一些预料库中学习到的词嵌入向量，会发现学习到的词向量存在下面一些具有性别、种族等偏见，这反映了人们在历史的写作中存在的这种社会偏见：



消除偏见的方法：

- 定义偏见的方向：如性别
- 对大量性别相对的词汇进行相减并求平均：$e_{he} - e_{she}$、$e_{male} - e_{female}$、$\cdots$；
- 通过平均后的向量，则可以得到一个或多个偏见趋势相关的维度，以及大量不相关的维度；
- 中和化：对每一个定义不明确的词汇，进行偏见的处理，如像 doctor、babysitter 这类词；通过减小这些词汇在得到的偏见趋势维度上值的大小；
- 均衡：将如 grandmother 和 grandfather 这种对称词对调整至babysitter这类词汇平衡的位置上，使babysitter这类词汇处于一个中立的位置，进而消除偏见。



更多算法的细节可以查阅图中的文献。

编辑于 2018-03-05 13:19

深度学习（Deep Learning） 人工智能 自然语言处理