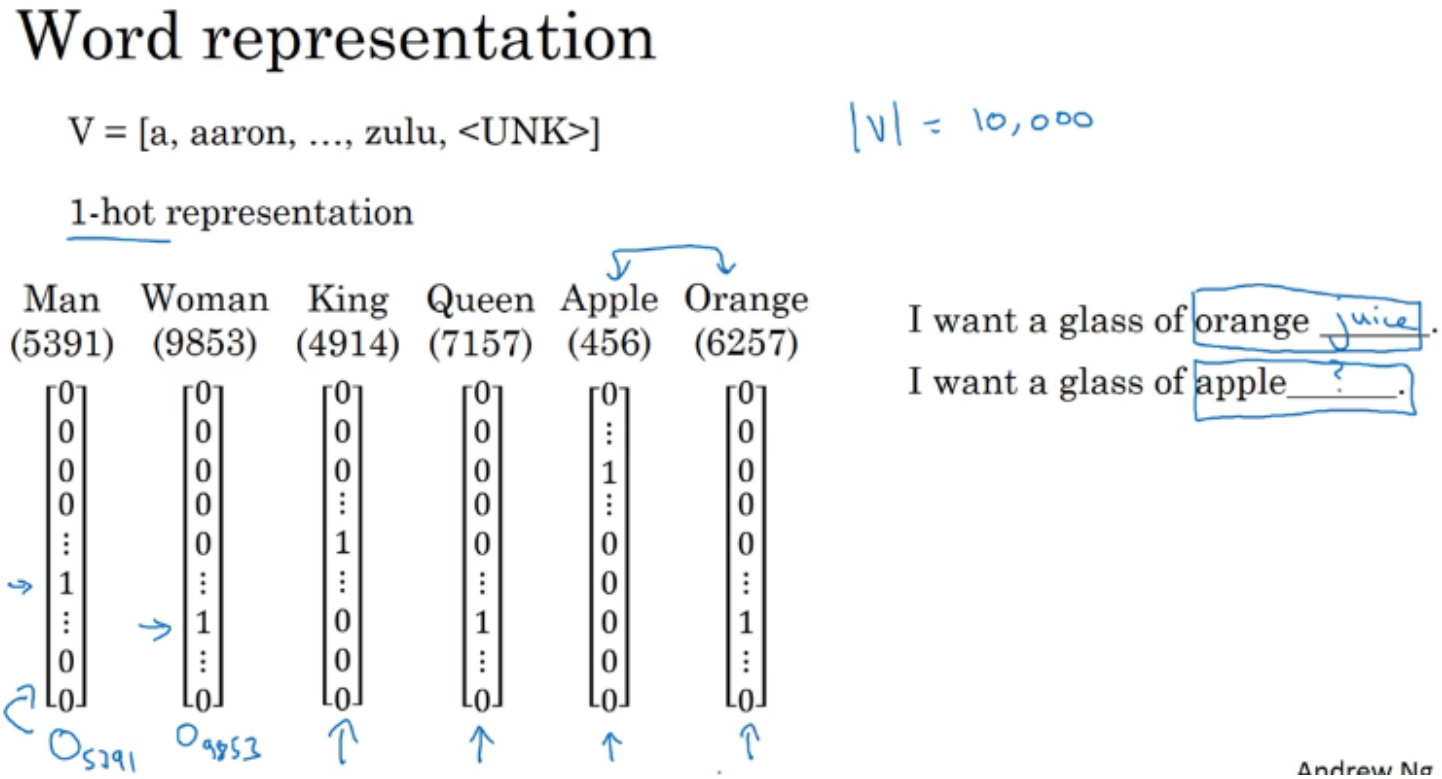
1. 词汇表征

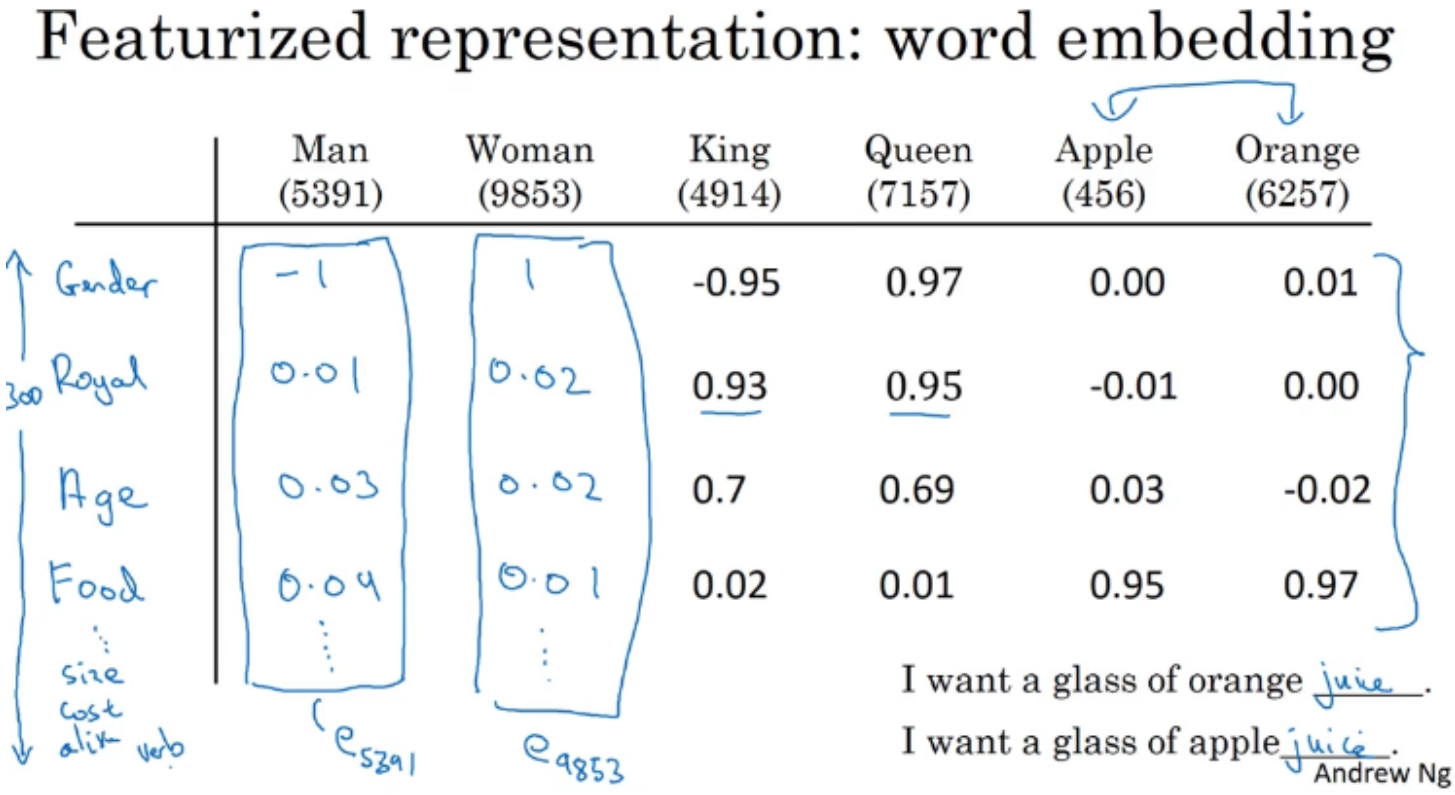
1.1 One-hot向量

之前的课程使用one-hot向量来表示词，对应单词的位置用1表示，其余位置为0。这种方法将每个词孤立，使得模型对于相关词的泛化能力不强。而且每个词向量之间的距离都一样，乘积为0，无法获取词之间的相关性和关联性。

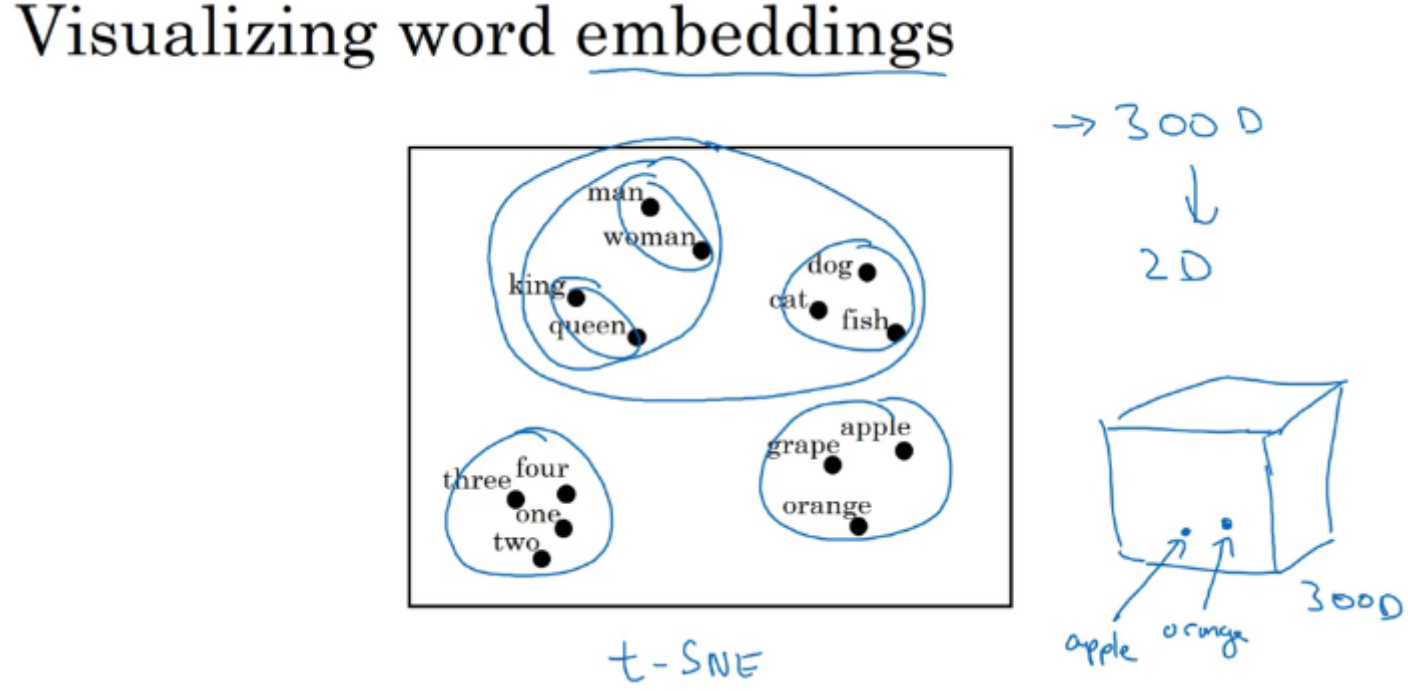


1.2 特征表征：词嵌入

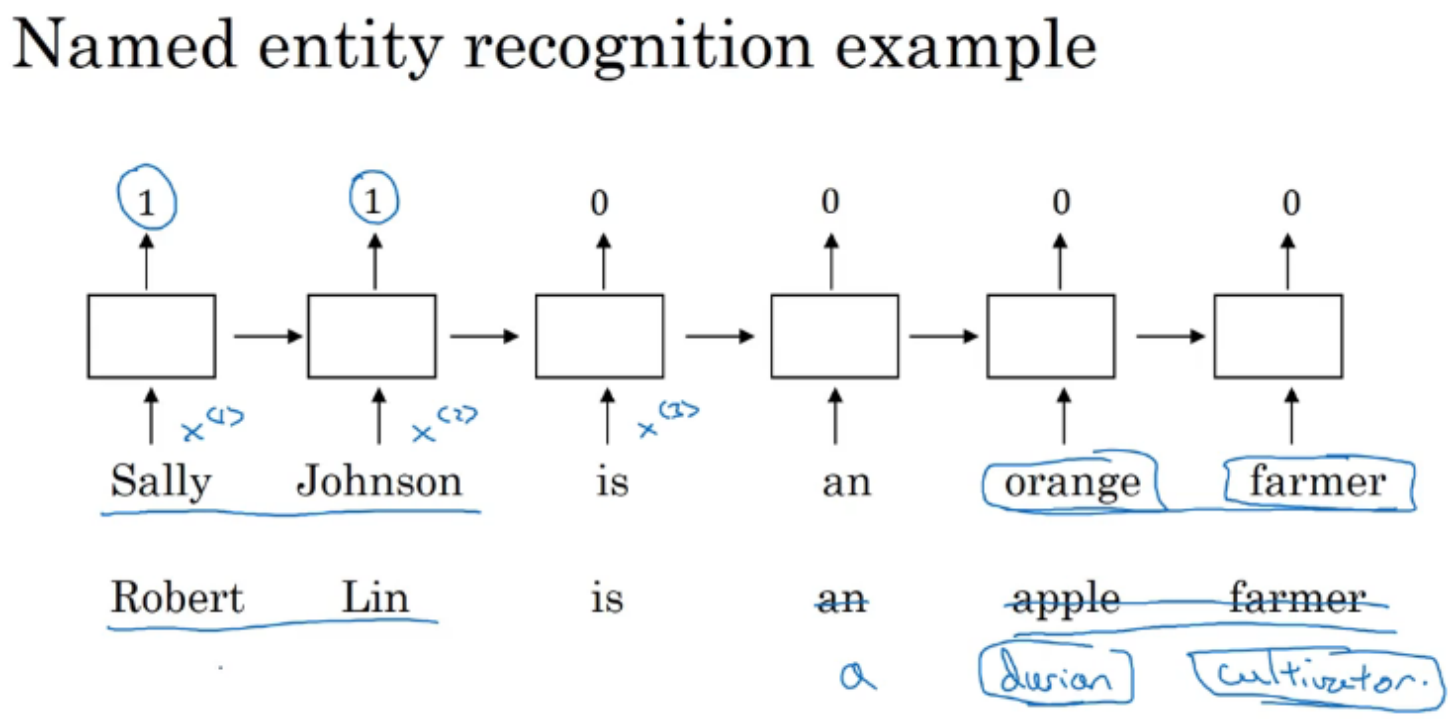
用不同的特征对每个词汇进行表征，这使得词之间的相似性更容易显示出来。



使用t-SNE算法将高维词汇映射到二维空间对词向量进行可视化，可以发现相似的词总是聚在一起。



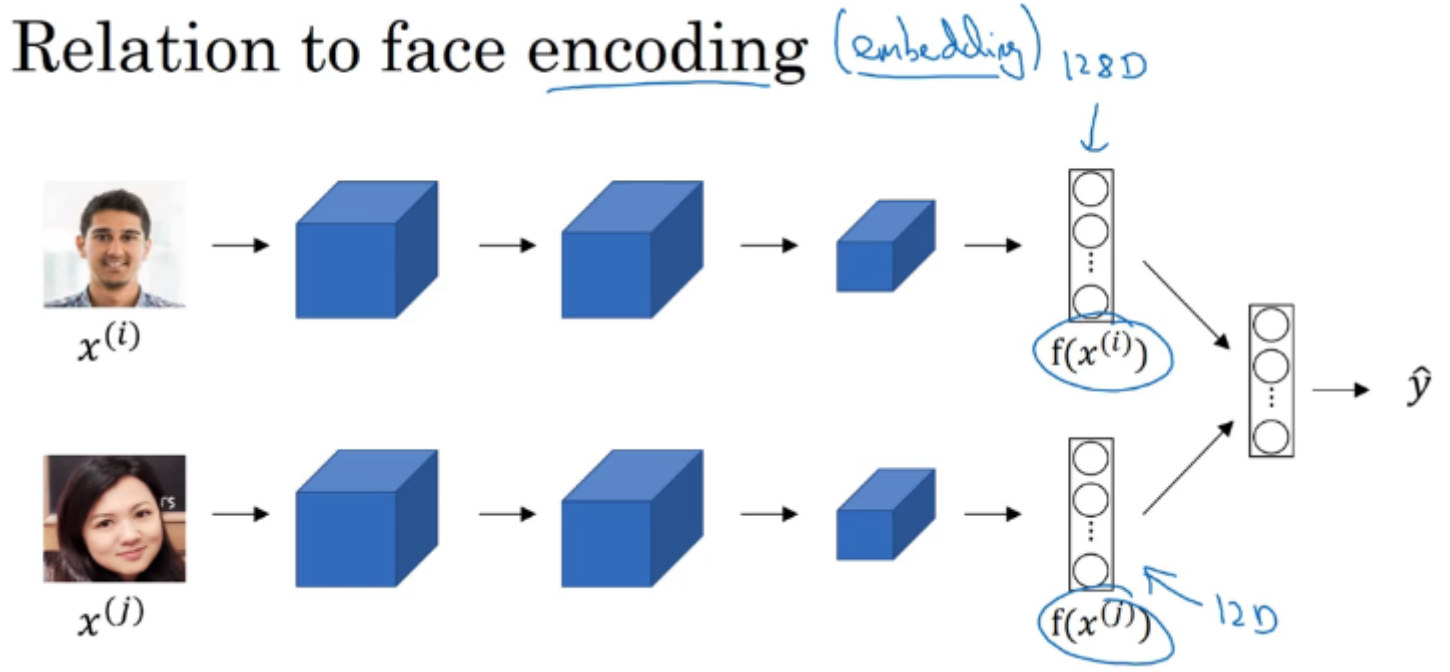
2. 使用词嵌入



如果数据集较小，可能不包含durain和cultivator这样的词汇，那么久很难从包含这两个词汇的句子中识别名字实体。但如果从网上获取了一个word embedding，它将告诉我们durain cultivator也是一个人，那么我们可以从少量的训练集中归纳出没有见过的词汇中的名字实体。

有了word embedding，我们可以使用迁移学习从网上的大量无标签文本中学习到的知识，应用到少量文本训练集的任务中。

Word embedding和face encoding有着奇妙的联系。在face recognition中，我们会将不同人脸图片编码成不同的向量，在识别的过程中使用编码来进行比对识别。



但对于face recognition，我们可以将任意人脸图片输入到网络中得到对应的人脸编码。而在word embedding中所有词汇的编码是在一个固定的词汇表中进行学习的。

3. Word embedding的特性

3.1 类比推理

通过不同词向量的相减计算可以发现不同词之间的类比关系。

Analogies

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	−1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

e_{5391} e_{man} e_{woman}

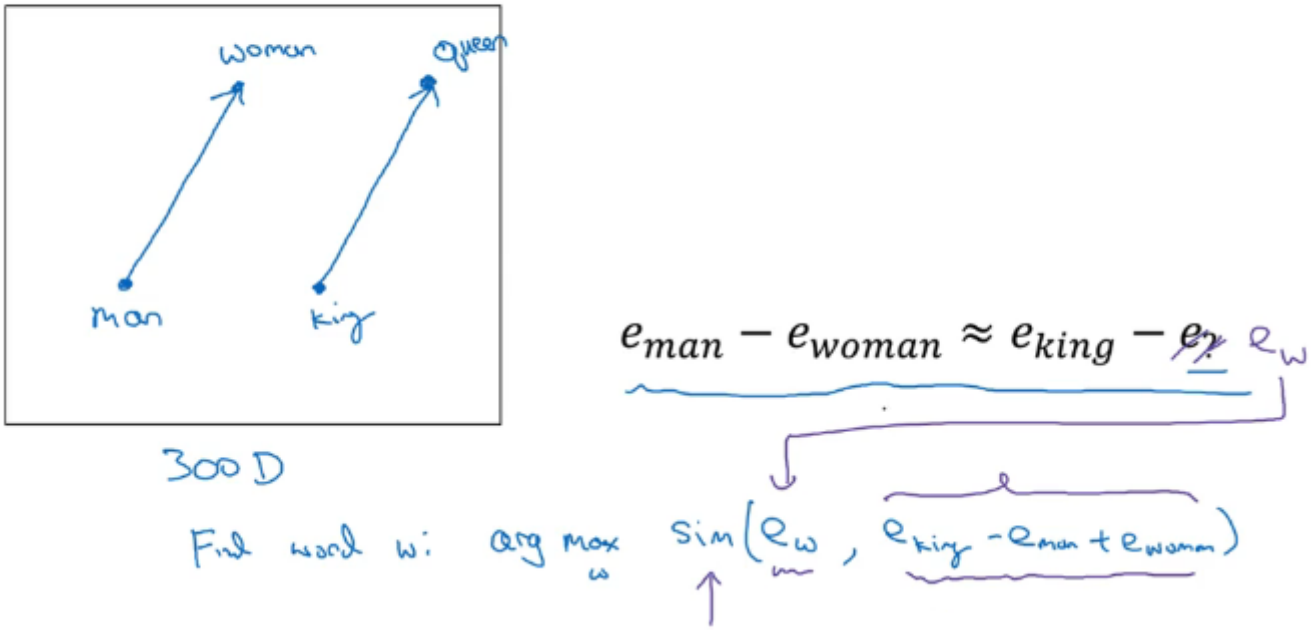
$e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Man → Woman as King → ? Queen

$e_{king} - e_{queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

计算词之间的相似度，实际上也就是寻找词向量在各维度的距离相似度。

Analogies using word vectors



寻找$e_?$，相当于寻找虾米那两个结果之间的最大相似度：

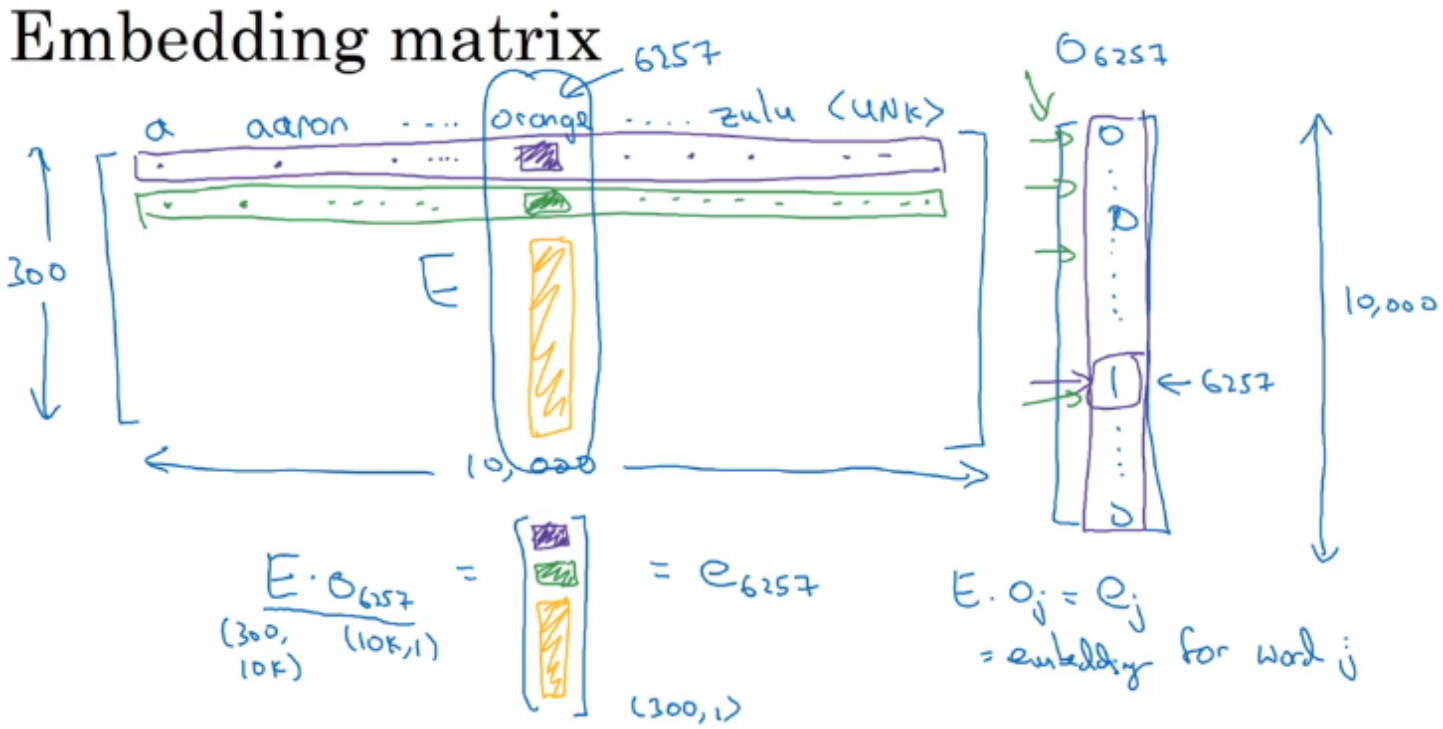
$$\arg\max_{?} sim(e_?, e_{king} - e_{man} + e_{woman})$$

3.2 相似度函数

- 余弦相似度：$sim(u, v) = \frac{u^T v}{||u||_2, ||v||_2}$
- 欧氏距离：$||u - v||^2$

4. 嵌入矩阵

对一个词汇表学习词嵌入模型时，实际上就是要学习这个词汇表对应的一个嵌入矩阵E。将学习好的嵌入矩阵与单词的one-hot向量相乘，得到该词的embedding。



5. 学习word embedding

人们一开始使用的学习word embedding的算法比较复杂，但随着时间的推移，算法变得越简单而且越有效。

5.1 早期的学习算法

- 假设词汇表的大小为10000，模型的参数为嵌入矩阵E，隐藏层和softmax层的参数$w^{[i]}$和$b^{[i]}$。
- 将单词的one-hot向量与嵌入矩阵相乘，得到单词的embedding (300)。
- 利用历史窗口 (4) 控制影响预测结果的单词数量，将窗口中的单词的embedding堆叠输入 (4*300=1200) 到神经网络中。
- 利用softmax输出结果（词汇表中每个单词的概率，10000）。
- 利用反向传播进行梯度下降训练参数。

5.2 其他上下文和目标词对

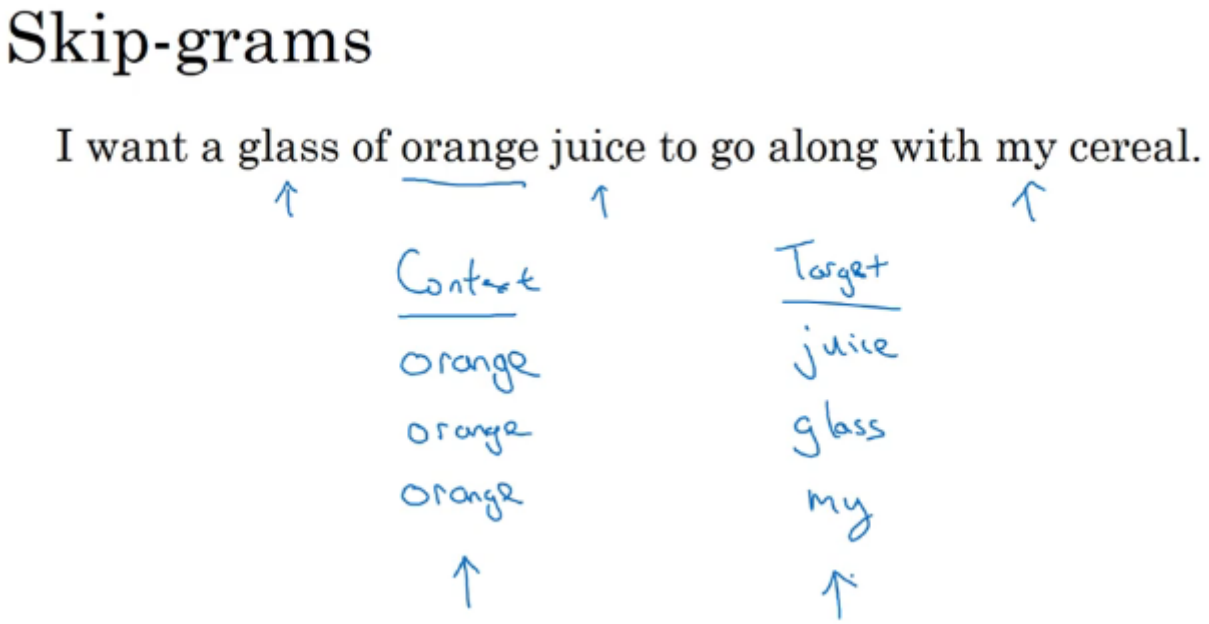
要预测的单词称为目标单词，通过一些上下文预测出来。

- 选取目标单词之前的几个词。
- 选取目标单词前后的几个词。
- 选取目标单词前的一个词。
- 选取目标单词附近的一个词（Skip-Gram）。

6. Word2Vec

6.1 Skip-grams

抽取context和target配对，构造一个监督学习问题。context不一定是离target最近的单词，而是随机选择一个词作为context，同时在context的一定距离内随机选择另一个单词作为target。构造这样一个监督学习问题的目的并不是为了解决监督学习问题本身，而是想使用这个问题来学习一个更好的word embedding模型。

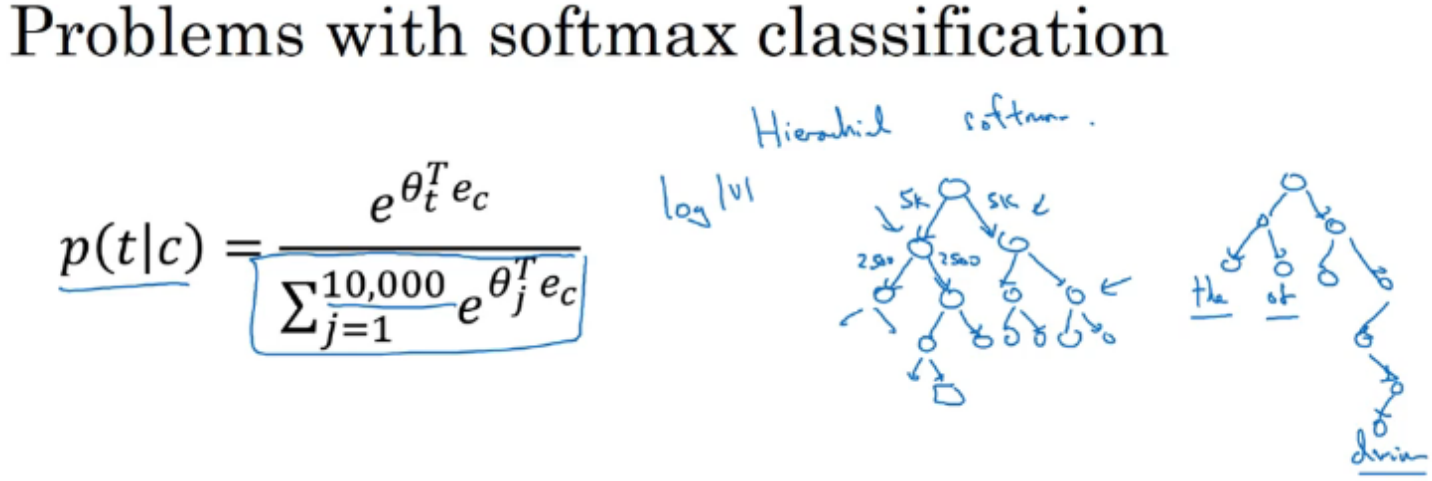


6.2 模型

- 词汇表：Vocab size: $10000k$。
- 构建context到target的映射关系。
- $o_c \rightarrow E \rightarrow e_c \rightarrow softmax \rightarrow \hat{y}$
- Softmax: $p(t|c) = \frac{e^{\theta_i^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$
- Loss: $L(\hat{y}, y) = -\sum_{i=1}^{10000} y_i log\hat{y}_i$

6.3 Softmax存在的问题

- 需要对10000个（甚至更多）词汇进行计算，计算量庞大。
- 简化方法：使用分级softmax。



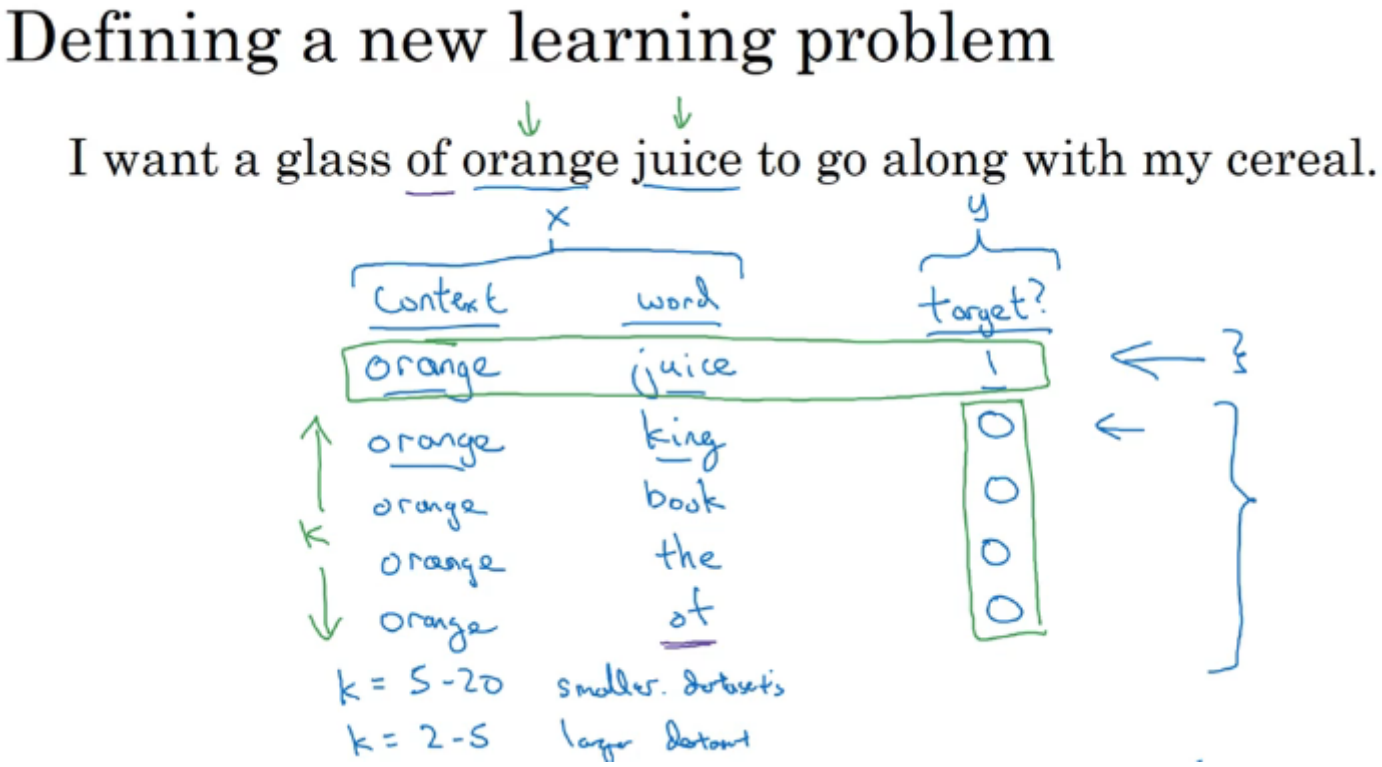
6.4 如何采样context

- 对语料库均匀随机采样，使得the、of这些词会出现的相当频繁，导致context和target经常出现这些词汇。
- 我们实际采用的方法是使用不同的启发来平衡常见和不常见的词。

7. 负采样

7.1 定义新的学习问题

- 预测两个词之间是否为context-target词对，是的话为1，不是的话为0。
- 使用相同的context，随机选择k个不同的target，并对相应的词对进行正负样本的标记，生成training set。
- 小数据集：$k = 5 - 20$，大数据集：$k = 2 - 5$。
- 学习$x - y$的映射。

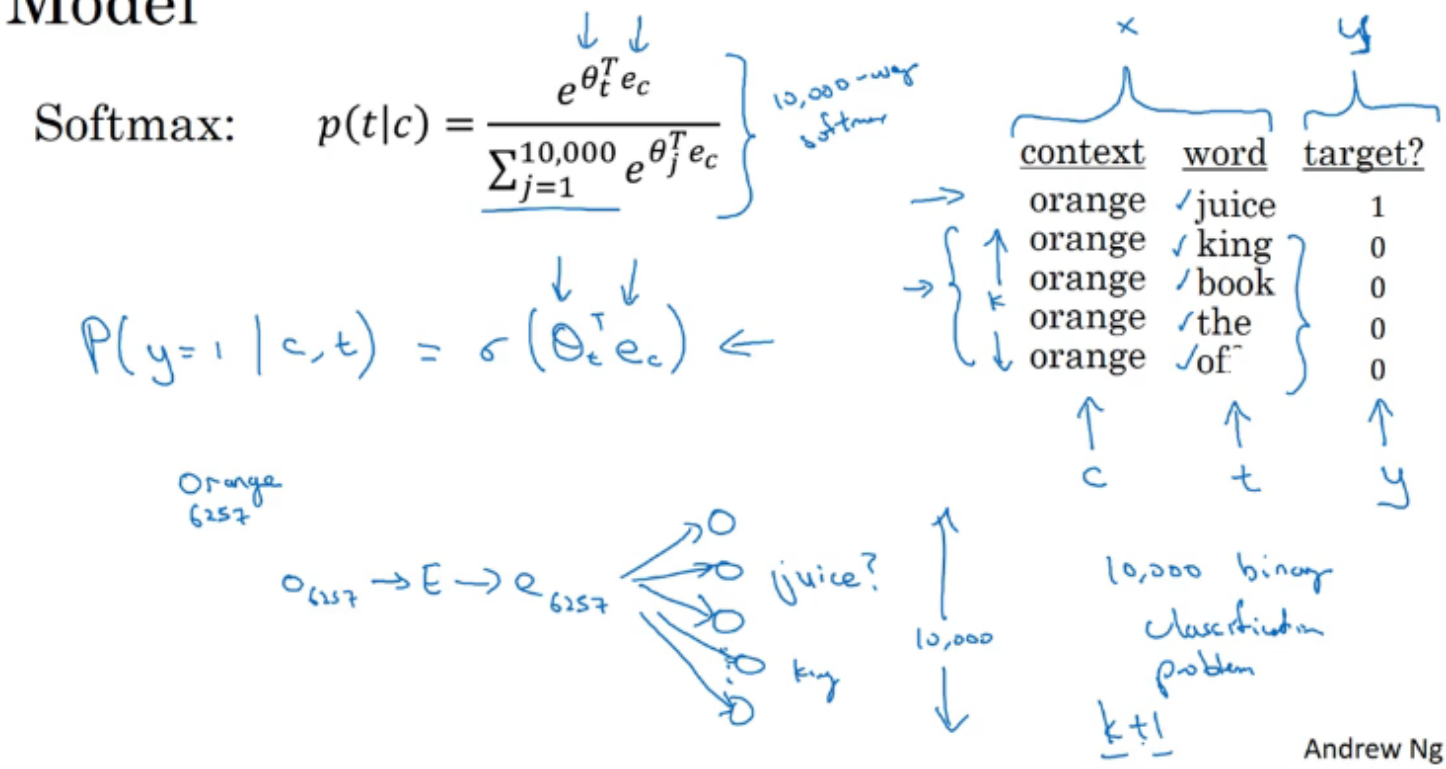


7.2 模型

使用logistic回归模型：

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

k=4时，每个正样本均有4个对应的负样本。对于每个context，我们有对应的5个分类器。



相比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的softmax，而是将其转化为词汇表大小个二分类问题。这使得迭代的成本大大降低。

7.3 如何选择负样本

选定context以及确定正样本后，需要选择k个负样本进行训练。

- 通过单词出现的频率进行采样，这使得the，of和and这些词的频率较高。
- 均匀随机抽取负样本，这对于英文单词的分布并没有什么代表性。
- $p(w_i) = \frac{f(w_i)^{\frac{3}{4}}}{\sum_{j=1}^{10000} f(w_j)^{\frac{3}{4}}}$。其中$f(w_i)$代表某个词的词频。

8. GloVe词向量

8.1 GloVe模型

- 定义一个X_{ij}，表示target i出现在context j中的次数。
- 优化目标：$minimize \sum_{i=1}^{10000} \sum_{j=1}^{10000} f(X_{ij})(\theta_i^T e_j + b_i + b_j = logX_{ij})^2$
- $f(X_{ij})$是加权项。当$X_{ij} = 0$时，$f(X_{ij}) = 0$。因为$X_{ij} = 0$时，$logX_{ij}$没有意义。另外$f(X_{ij})$对频繁词和不频繁词有着启发式的平衡作用。
- $\theta_i^T e_j$是需要学习的参数，在这个模型中这两个参数是对称的。可以一致地初始化θ和e。最后取二者的平均值作为结果：$e_w^{final} = \frac{e_w + \theta_w}{2}$。

8.2 词向量特征化

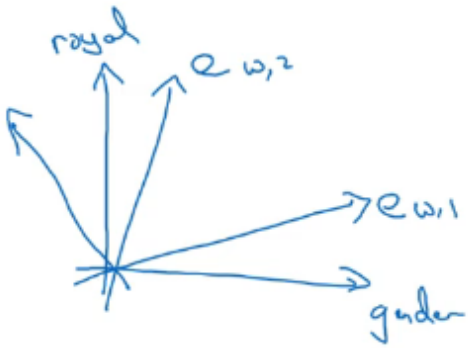
通过上面的算法得到的word embedding向量，我们无法保证词其每个独立分量是能够让我们理解的。但是每个分量和我们所想的特征是有关联的，其可能是一些我们能够理解的特征组合而构成的复合分量。使用上面的GloVe模型，从线性代数的角度解释：

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

A项可能构成任意的分量组合。

A note on the featurization view of word embeddings

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	
Gender	−1	1	-0.95	0.97	←
Royal	0.01	0.02	0.93	0.95	←
Age	0.03	0.02	0.70	0.69	←
Food	0.09	0.01	0.02	0.01	←



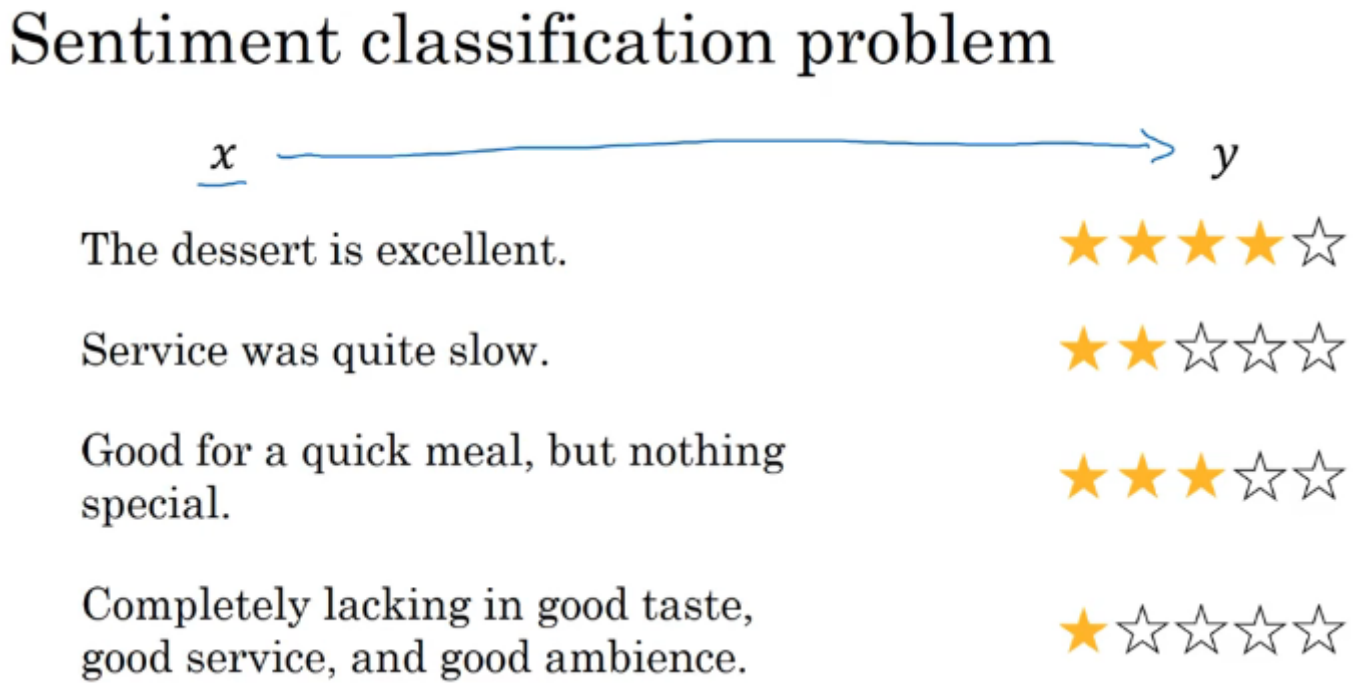
$$\text{minimize} \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij})(\theta_i^T e_j + b_i - b_j' - \log X_{ij})^2$$

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

Andrew Ng

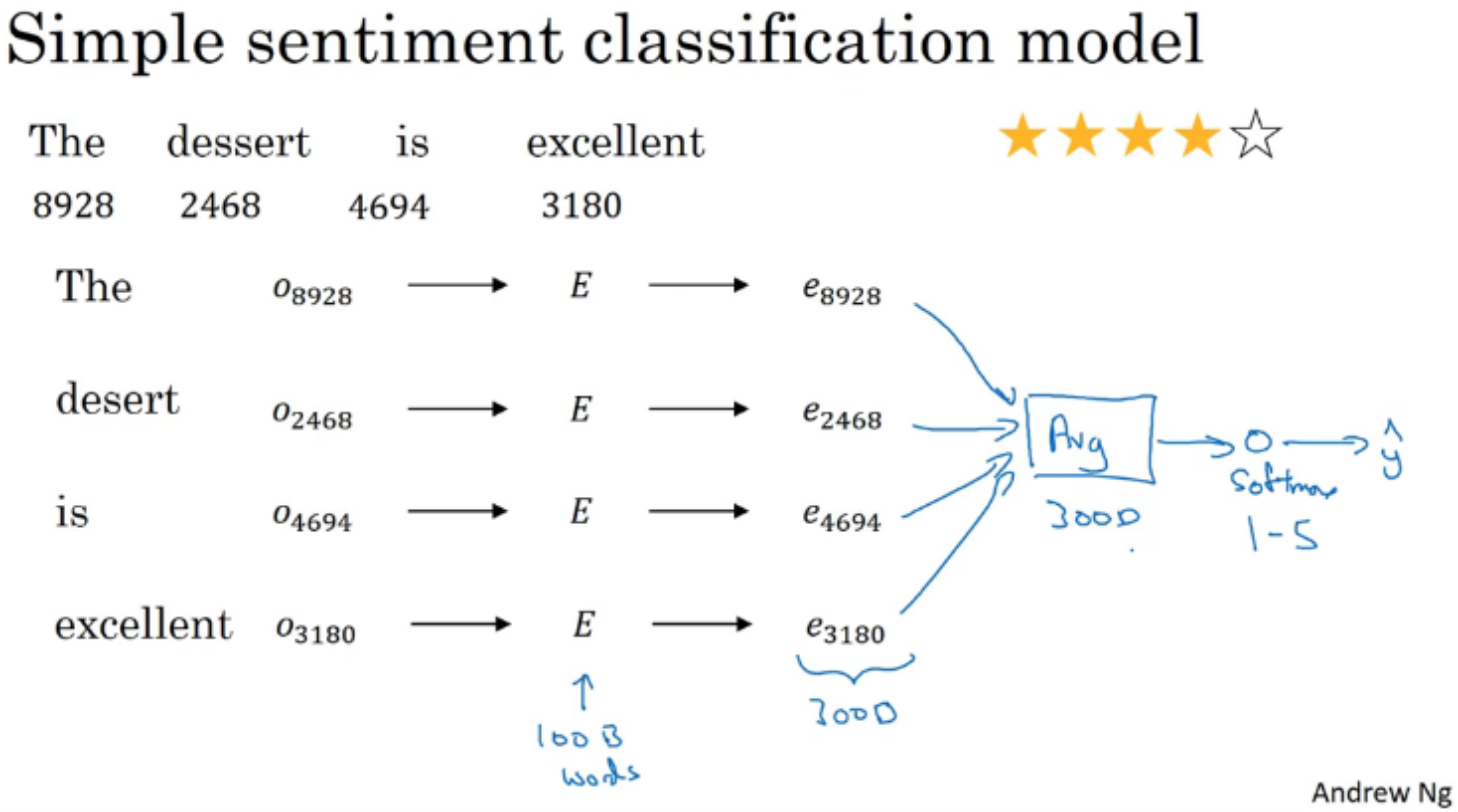
9. 情感分类

情感分类就是通过一段文本来分析这个人是否喜欢他们正在讨论的东西，是NLP中最重要的模块之一。

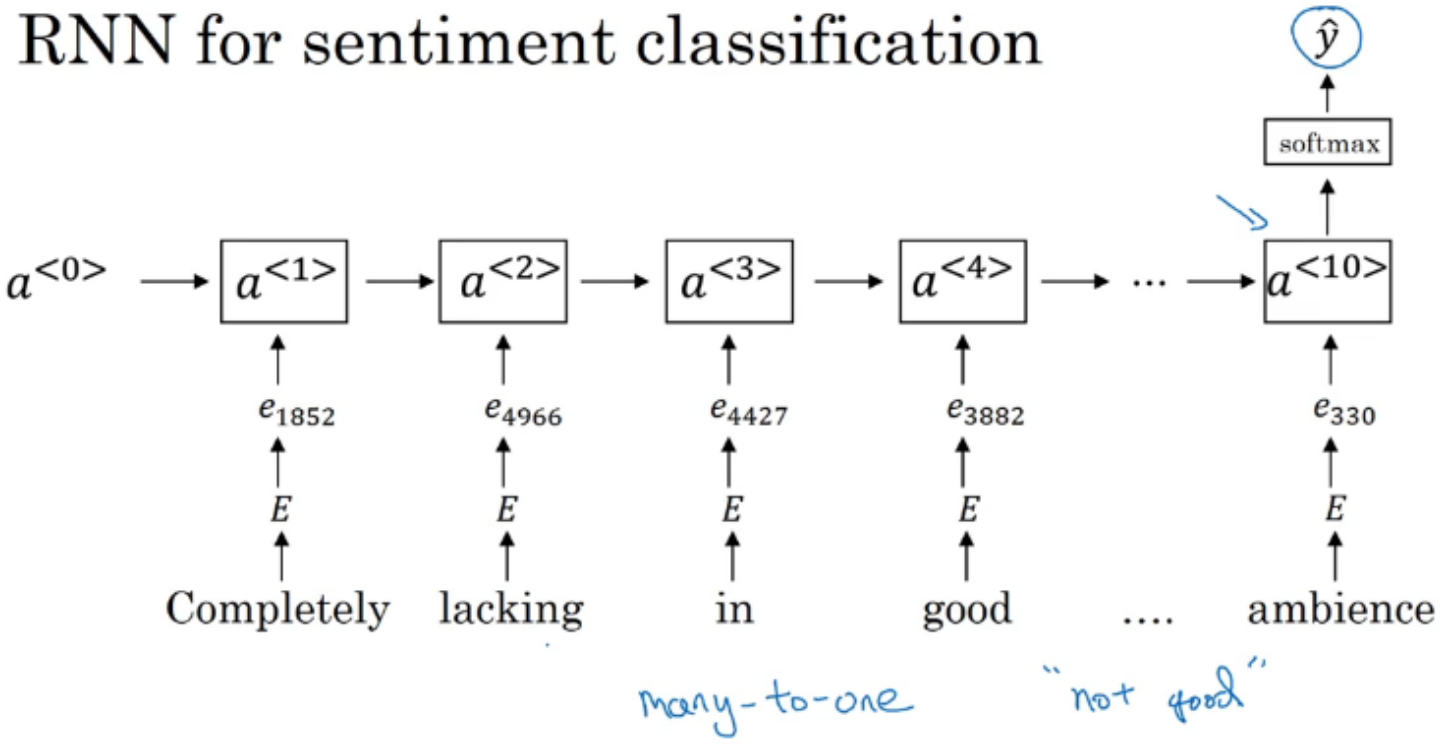


情感分类存在的问题就是数据集太小，缺乏训练样本。但是有了word embedding后，可以带来很好的效果。

一种方法是获取训练好的word embedding矩阵E，计算每个词的word embedding向量，对所有的向量求和或者平均，再把这个结果输入到softmax中，得到最后的输出。缺点就是没有考虑词序，多数的积极词汇核能削弱消极词汇的影响，造成错误的预测。



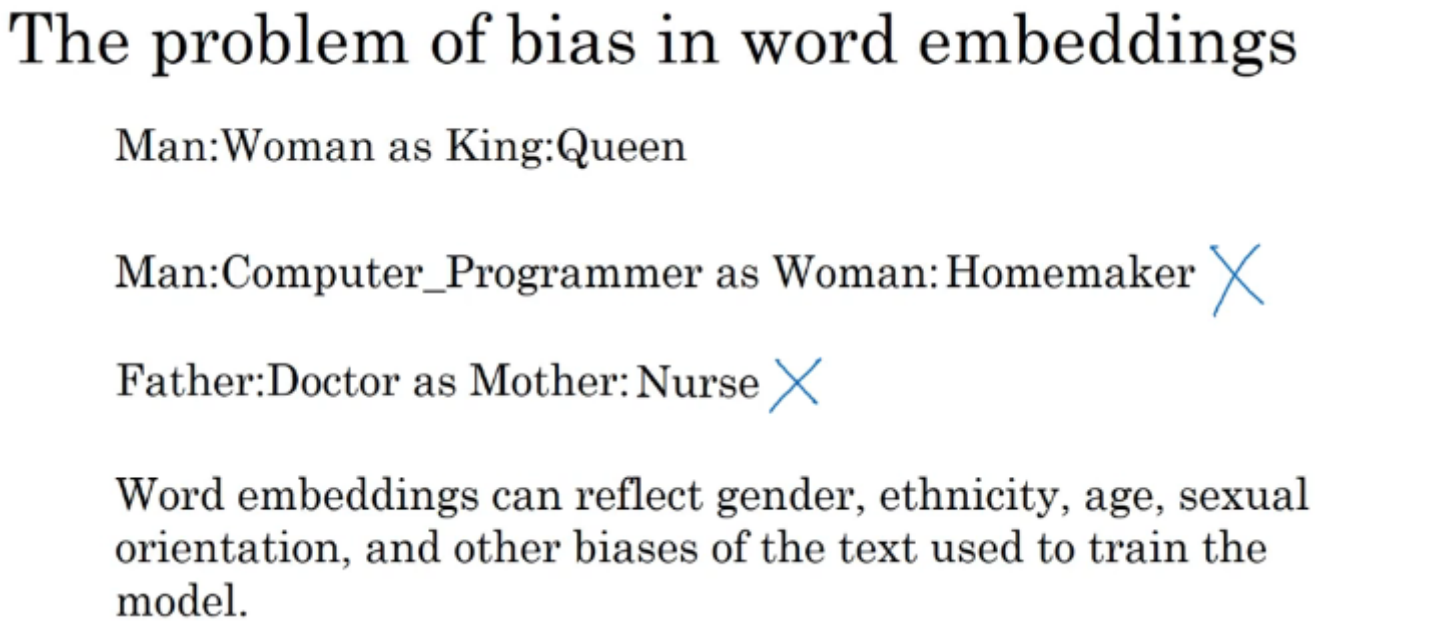
另一种方法是使用RNN模型。将word embedding向量输入到many-to-one的RNN模型中。



10. Word embedding消除bias

机器学习或者人工智能算法已经被应用到制订重要的决策中，我们需要尽可能保证其不受非预期形式的bias的影响，如性别、种族歧视等。

10.1 目前存在的问题



10.2 消除bias的方法

- 定义bias的方向：比如性别。对性别词汇进行相减求平均：$e_{he} - e_{she} \cdots$。通过平均化的向量，得到一个或多个bias趋势相关的维度以及大量不相关的维度。
- 中和化（Neutralize）：对于每个不明确定义的词汇进行bias的清除，如doctor，babysitter等。减少它们在bias方向上值的大小。
- 均衡（Equalize）：比如将grandmother和grandfather这些词汇调整至babysitter这类词汇平衡的位置上，使babysitter这些词汇处于一个中立的位置，进而消除bias。