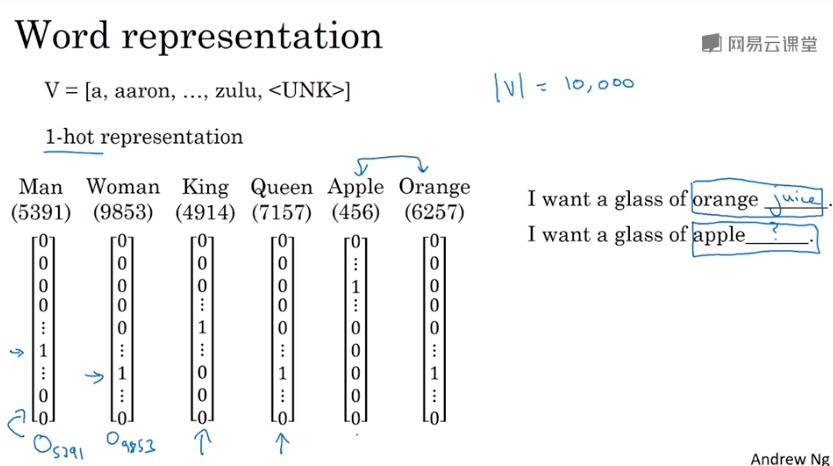
# 第二章 自然语言处理与词嵌入

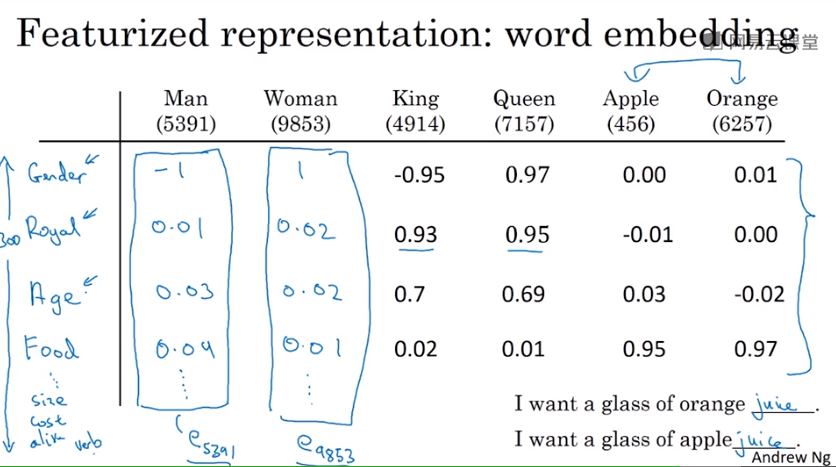
## 2.1 词汇表征

上一节学习了RNN的GRU单元和LSTM单元，这一节学习如何把这些运用到自然语言处理NLP问题上，关键概念：词嵌入（word embeddings），这是语言表示的一种形式，可以让算法自动理解一些类似的词，比如男人和女人。即使标记的训练集相对较小，有了词嵌入的概念就能应用NLP了。最后会消除词嵌入的偏差，就是去除不想要的特性或者其他类型的偏差。

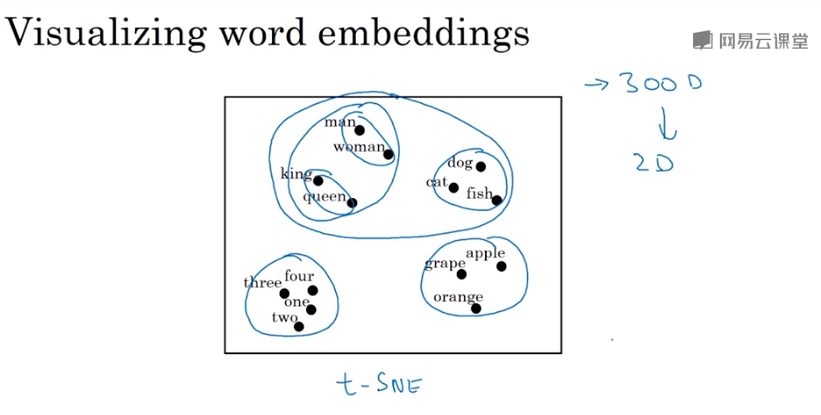
单词表示：起初都是用词汇表one-hot向量来表示词，比如man这个单词在词汇表的5391处，用来表示这个one-hot向量。这种表示方法的缺点是把每个词孤立起来，使得算法对相关词的泛化能力不强。比如橙子和苹果是一种东西，算法已经学习到一个短语，橙汁，却不知道苹果汁也是一类。因为orange和apple的one-hot向量内积都是0，并且向量间的距离都一样，所以相同类型词之间得泛化学习能力很弱。



所以换一种表示方法，特征化的表示featurized represention，学习这些值的特征或者数值化，也就是说从很多特性上给这些词进行打分，比如，年龄，性别，是否高贵，食物类，大小，花费等等等，可以想很多的特征，假设有300个特征，那么也就有300个数字分别对应与这300个特征的符合程度，组成一个300维的向量，比如man这个词用来表示，表示woman这个词。使用这种词表示方法，那么apple和orange的两个向量就会很相似，但是也会有不同，比如颜色等特征的数值。接下来要学的特征表示方法不会像表里的那些特征那么好理解，但是能使算法高效的发现苹果和橙子会比国王与橙子更加相似。

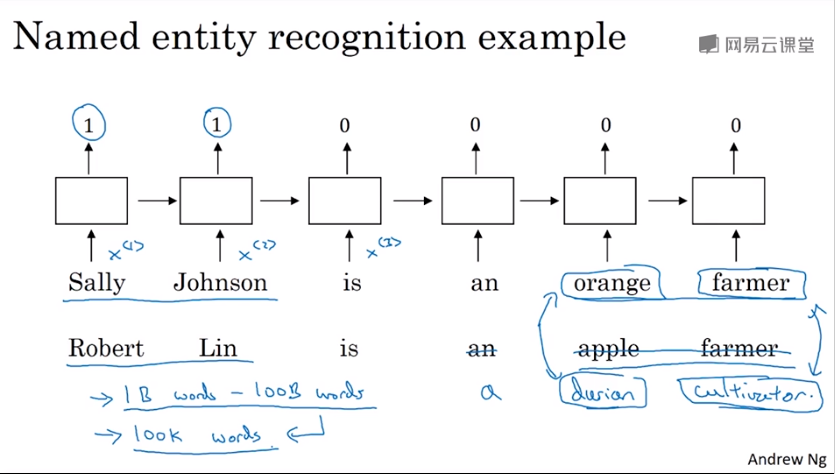


如果可以学习到一个300维的特征向量或者说300维的词嵌入，通常可以做的就是把这300维的数据嵌入到一个二维空间内，就可以进行可视化了，常用的可视化算法是t-SNE算法，这样可以可视化高维的数据。这种词嵌入算法对于相近的概念，学到的特征也比较类似，在对这些概念可视化的时候，这些概念也会比较相似，最终把它们映射为相似的特征向量。这种表示方法用的是在300维空间里的特征表示，这就叫做嵌入，原因就是想象一个300维空间，这个词就被嵌入到300维空间的一个点上，为了可视化使用t-SNE算法，把这个空间映射到低维空间，可以得到一个二维图像进行观察。



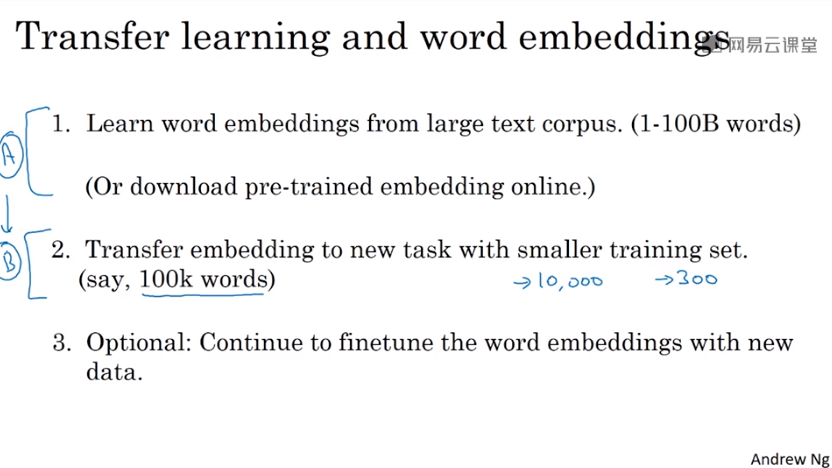
## 2.2 使用词嵌入

这节学习如何把词的特征化表示应用到NLP中，首先继续使用命名实体识别的例子，找出人名，比如在句子Sally Johnson is an orange farmer中能知道Sally Johnson是一个人名，而不是公司名，是因为知道farmer一定是一个人。所以经过上节学习，用词嵌入作为输入来训练好的模型。如果看到一个新的输入，Robert Lin is an apple farmer，因为apple与orange很相近，你的算法也就很容易知道后面也是一个人名。一种有趣的情况是，如果XX是一个榴莲培育家，并且如果在命名实体识别任务中，只有一个很小的标记的训练集，训练集里面没有durian和cultivation这两个词，但是如果你有很好的词嵌入，知道榴莲是水果，培育家是人，那么算法还是可以学到XX也是人名。词嵌入可以达成这种效果原因就是学习词嵌入算法会考察非常大的文本集，一亿甚至是100亿的单词，所以存在没有标签的文本的一个很大的训练集，考察大量的无标签文本，可以发现orange apple 与durian相似。 因此词嵌入算法可以把同类事物聚集在一起，通过读取大量的互联网文本，发现orange和durian都是水果，就可以应用到你的命名实体识别任务当中了。尽管只有一个很小的训练集，但是可以使用迁移学习，把从互联网上免费获得的大量的无标签文本中学习到的知识，能够区分类别的迁移到另一个任务中。虽然图中画的是单向的RNN，但是在命名实体识别任务中，应该使用双向RNN。

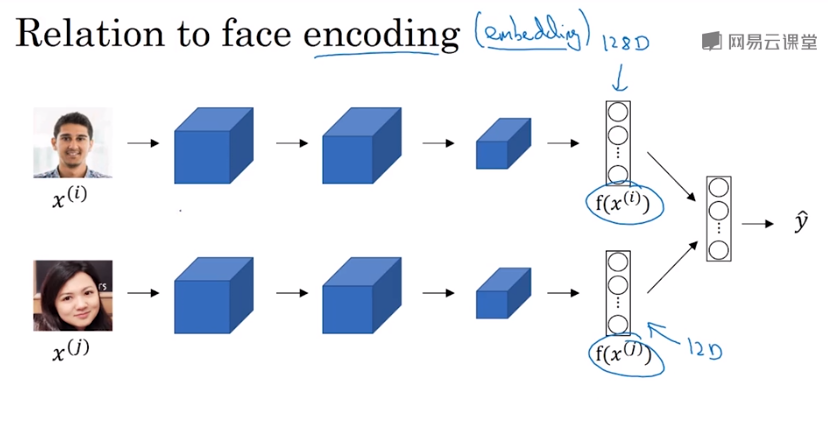


总结用词嵌入做迁移学习的步骤：

1. 先从大量的文本集中学习词嵌入，一个非常大的文本集，或者下载网上提前训练好的词嵌入模型。
2. 用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，比如用这个300维的词嵌入来表示所有的单词，好处在于可以用更低维度的特征向量代替原来的10000维的one-hot向量
3. 在新的任务上训练模型时，在你的命名实体识别任务上，可以选择要不要用新的数据进行微调词嵌入。当然这一步是在当第2步中有很大的数据集时才会用。

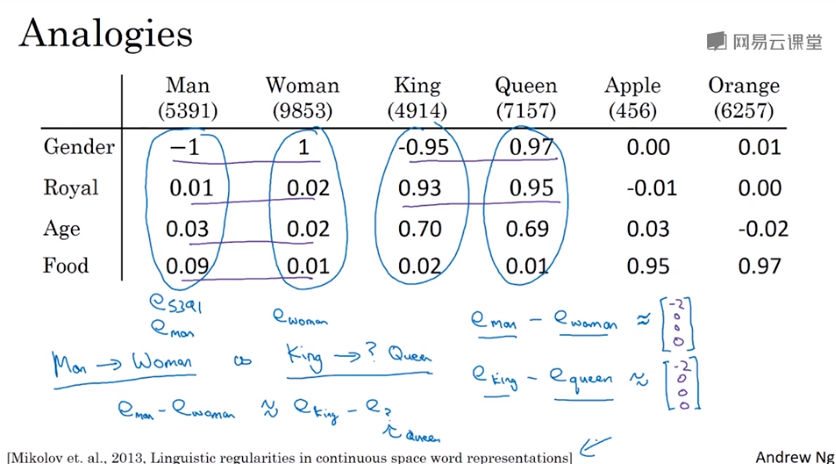


在数据集量越小的时候，词嵌入的作用越明显，广泛用于NLP领域。它已经用在命名实体识别，文本摘要，文本解析，指代消解，这些都是标准的NLP文物，词嵌入在语言模型，机器翻译领域用得少一点，，尤其是做语言模型或者机器翻译任务这些任务有大量的数据时。所以和在其他迁移学习中的情形一样，如果从某一任务A迁移到某个任务B，只有A中有大量数据，而B数据少时，迁移才有用。所以对于大多数NLP是有用的，但是对于一些语言模型和机器翻译则用处不大。

词嵌入和人脸编码之间有奇妙的关系，在人脸识别中，训练了一个Siamese网络结构，这个网络会学习不同人脸的一个128维表示，然后通过编码结果来判断两个图片是不是同一个人脸。词嵌入和这个差不多，人脸识别中喜欢用编码这个词，不同的是，人脸识别中训练一个网络任给一个人脸照片甚至是没见过的照片，神经网络都会计算出一个编码结果。而学习词嵌入中则是有一个固定的词汇表，比如10000个单词就有到个对应的表示，学习一个固定的编码，对于每一个词汇表的单词都有一个固定的嵌入，对于没出现过的单词就记为未知。

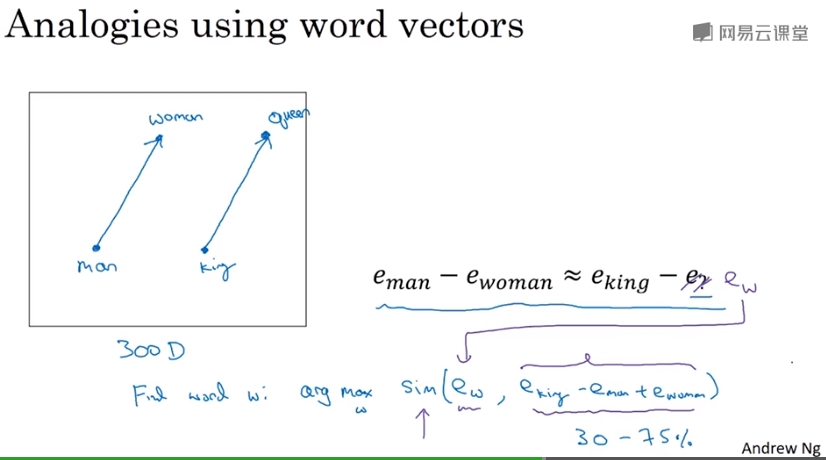
## 2.3 词嵌入的特性

词嵌入除了可以帮助构建NLP应用，还可以帮助实现类比推理analogy reasoning。尽管类比推理可能不是NLP应用中重要的存在，但是可以帮助人们理解词嵌入到底做了什么以及能够做什么。下图是一些词嵌入可以捕捉的词的特征化表示，假设明确man->woman，那么king应该对应什么？queen能否有一种算法可以自动推导出这种关系，下面就是实现方法。



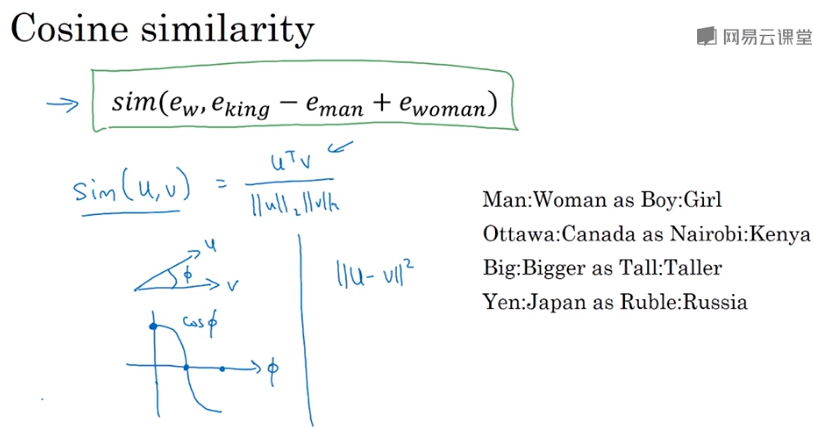
首先用四维向量表示man，假设现在使用的四维嵌入向量，而不是一般的50-1000维的向量，这些向量有个有趣的特性就是，将两个向量做减法

可以发现woman和man，以及king和queen的差距主要是性别上的差异，所以得出这种类比推理的结论的方法就是当算法被问及man对woman，那么king对什么时，算法要做的就是使用，然后找出一个向量，也就是一个词，使得，这是词嵌入领域影响力最为惊人和显著的成果之一。



如何把这种思想写成算法，也就是说需要找到这么一个词w，使得sim（），使得相似的程度最大。有关于计算单词w和右边这几项之间得相似度函数，通过函数找到一个最大的结果，如果结果理想，就会得到queen。这种方法确实有效，如果学习一些词嵌入，通过算法来找到使得相似度最大化的单词w，确实可以得到完全正确的答案，不过这取决于过程的细节。查看论文，这种方法的准确率大概只有30%—70%。之前说到使用t-SNE算法进行可视化，t-SNE做的是使用一种非线性的方式映射到二维平面上，所以其中算法很复杂并且非线性。在知道t-SNE映射之后，不能总是期望使等式成立的关系会向平面两组数据距离相等，呈平行四边形。

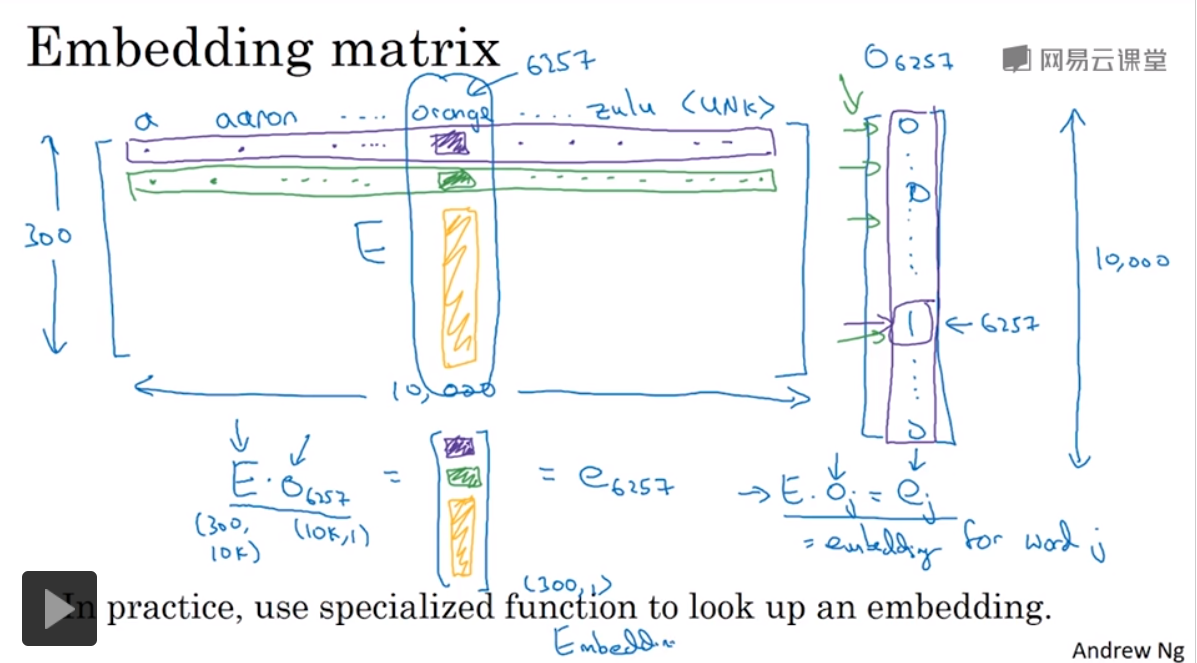
一个最常用的相似度函数，余弦相似度，假设在向量u和v中定义相似度，，分母表示的是两个向量的欧氏距离，如果两个向量相似，内积分子就会很大。叫做余弦相似度就是因为，其实就是计算两个向量的夹角的余弦值。当夹角为0时，余弦值最大。你也还可以使用平方距离和欧式距离来表示，，从学术上说，比起测量相似度，这个函数更容易测量的是相异度，所以需要对其取负，函数才能正常工作。余弦相似度还是使用的更多，这两个的区别是，它们对u和v之间的距离标准化的方式不同。



词嵌入一个显著特点就是可以学习类比关系的一般性，除了性别，还比如说某个城市的首都等，比较级的表示等，这些东西都能够学习，只要你在大型的文本语料库上实现一个词嵌入学习算法，只要从足够大的语料库中进行学习，就能自主发现这些模式。

## 2.4 嵌入矩阵

接下来将学习词嵌入这一问题具体化，当你运用算法来学习词嵌入时，实际上是学习一个嵌入矩阵E。假设字典有10000个单词，要做的就是学习一个嵌入矩阵E，将是一个（300,10000）的矩阵，这个矩阵的每一列都表示了一个词。如果对于某个词如第6257个词orange，用one-hot表示是，如果用E乘，最后的结果是个300维的，其实就是E里面orange这一列的内容，即，所以表示的是第j个单词的嵌入向量。这一节目标就是知道我们要学一个嵌入矩阵E，先随机初始化，再使用梯度下降法来学习这个300x10000的矩阵中的各个参数，以及最后得到某个词的嵌入向量。在得到这个时，使用E与one-hot相乘的方式计算量比较大，高维并且很多都是0，所以需要一个函数单独查找矩阵E的某列。例如Keras中就有嵌入层，用这个嵌入层更有效的从嵌入矩阵中提取出你需要的列，而不是对矩阵进行很慢很复杂的乘法运算。

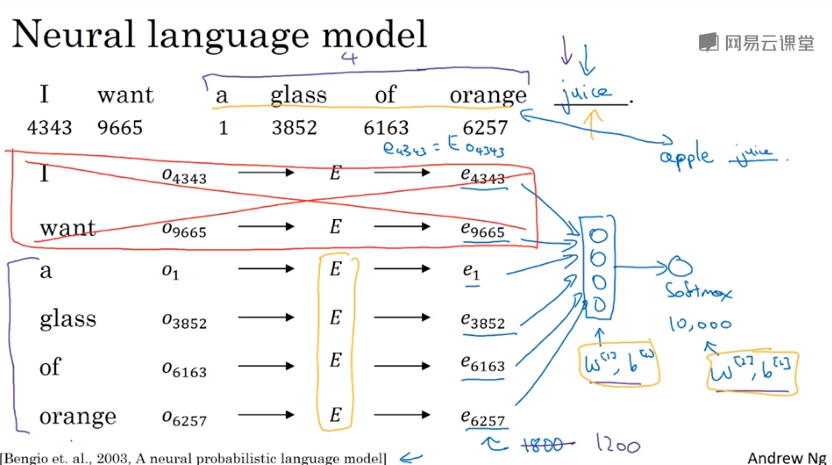


## 2.5 学习词嵌入

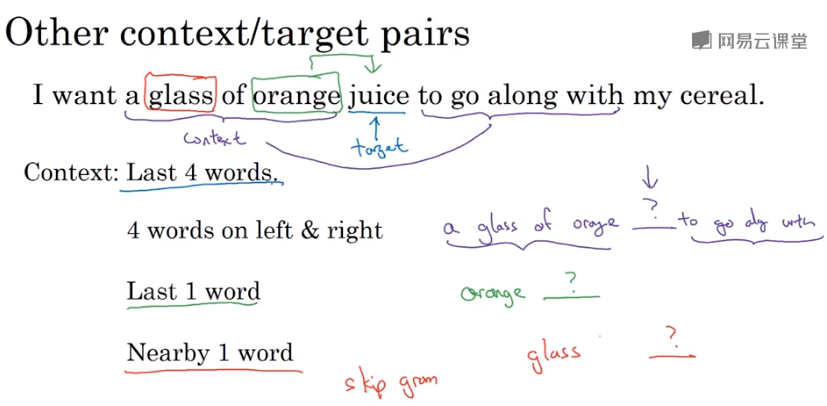
这节学习一些具体的算法来学习词嵌入，也就是学习矩阵E，在深度学习运用到这方面前，以前使用的算法比较复杂，但随着时间推移，研究者们不断发现能用更简单的算法来达到一样好的效果，尤其是在数据集很大的情况下。先从复杂的算法开始学，这样更容易对算法的运作方式有直观了解，后面会对算法进行简化，发现一些简单的算法也能有好的效果。

假如构建一个语言模型，并且用神经网络来实现这个模型，需要网络做到比如在输入I want a glass of orange\_\_\_\_时，可以预测下一个词。实践证明，建立一个语言模型是学习词嵌入的好方法，下面介绍如何建立神经网络来预测序列中的下一个词。

首先对一句话的每个词都有one-hot向量，随机初始化一个矩阵E，通过可以得到每个词的嵌入向量，比如这里有6个词，然后就是6个300维的向量，将它们堆叠在一起，得到一个1800维的向量，再经过隐藏层，最后到达softma层来进行预测，预测10000种词的可能，最后的目标是十softmax能预测出来下一个词是juice，隐藏层和输出层都有自己的参数W和b。还有个超参数，表示的是一个固定历史窗口大小，比如总是只想预测给定的四个单词a glass of orange的下一个单词，这个窗口大小数4就是超参数，这就是如何适应很长或很短的句子，总是只看前4个单词，而不看更前面的那些没用的词。如果一直使用4的窗口大小，就意味着神经网络会输入一个4\*300=1200维的特征向量到隐藏层中以及后面的softmax层来预测输出。这个超参数可以任意选择，选择有很多种，用一个固定的窗口就意味着可以处理任意长度的句子，因此输入的维度总是固定的。所以这个模型的参数就是矩阵E，对所有的单词都是使用同一个矩阵E，而不是对应不同位置上的不同单词用不同的矩阵，并且神经网络隐藏层等上的权重等也是算法的参数，然后可以用反向传播来进行梯度下降，来最大化训练集的似然值，通过序列中给定的4个单词去重复地预测出语料库中下一个单词是什么。事实上通过这个算法可以很好地学习词嵌入，并且apple和orange会学到很相似的嵌入，这样做能够让算法更好地符合训练集。如果只用一个300维的特征向量来表示所有这些词，算法会发现要想最好地拟合数据集，就要使apple，orange，pear等都拥有相似的特征向量。这就是早期的学习词嵌入，学习这个矩阵E的算法之一。



概括这个算法推导出更加简单的算法，举一个更加复杂的例子来进行说明，I want to a glass of orange juice to go along with my cereal假设训练集中有这样一个更长的例子，预测单词juice，目标词，是通过一些上下文本例中是前4个单词推导出来的，如果目标是学习一个嵌入向量，研究人员已经尝试了很多不同类型的上下文，如果要建立语言模型，那么一般选取目标词之前的几个词作为上下文，但如果你的目标不是语言模型本身的话，可以选择其他的上下文。比如需要预测中间的填空词，那么就需要前后的几个单词输入到神经网络中，来预测中间的目标词，也可以用来学习词嵌入。或者想用一个更简单的上下文，只提供目标前面前一个词，将是不同的学习问题，可以构建神经网络，只把前面一个词的嵌入向量输入到神经网络中来预测该词的下一个词是什么。还有一个效果比较好的，就是附近的某一个单词，比如选出附近的glass这个单词作为上下文，这用的是一种Skip-Gram模型的思想，这种比起前面4个词的简单的多，但是这种算法依然效果很好。



研究者发现，如果你真想建立一个语言模型，用目标词的前几个单词作为上下文是常见做法，但是如果你的目标是学习词嵌入，就可以用其他类型的上下文。这节关于语言模型，一个机器学习问题，即输入一些上下文来预测目标词，学习了提出这些问题是怎样帮助学习词嵌入的，下节学习如何使用更简单的上下文和更简单的算法里建立上下文到目标词的映射，将会让你更好的学习词嵌入。

## 2.6 Word2vec

上一节学习的如何学习一个神经语言模型来得到更好的词嵌入，Word2vec是一种更简单而且计算时更加高效的方式来学习这种类型的嵌入。假设在训练集中给定一个这样的句子，I want to a glass of orange juice to go along with my cereal，在Skip-Gram模型中，要做的就是抽取上斜纹和目标词配对来构建一个监督学习问题。上下文不一定总是目标单词之前离得最近的四个单词或最近的n个单词，要做的是随机选一个词作为上下文词，比如orange，然后随机在一定间距内选另一个词，比如前后5个词或者10个词内，就在这个范围内选择目标词，可能正好选择了juice，正好是下一个词，也有可能选择到了前面的词，所以有很多配对可能。于是就构造一个监督学习问题，给定上下文词要求预测这个词正负10个词距或者5个词距内随机选择的某个目标词，显然这不是个非常简单的学习问题。因为在单词orange的正负10个间距内可能有很多不同的单词，但是构建这个监督学习问题并不是为了解决这个问题本身，而是使用这个问题来学到一个好的词嵌入模型，接下来是模型的细节。

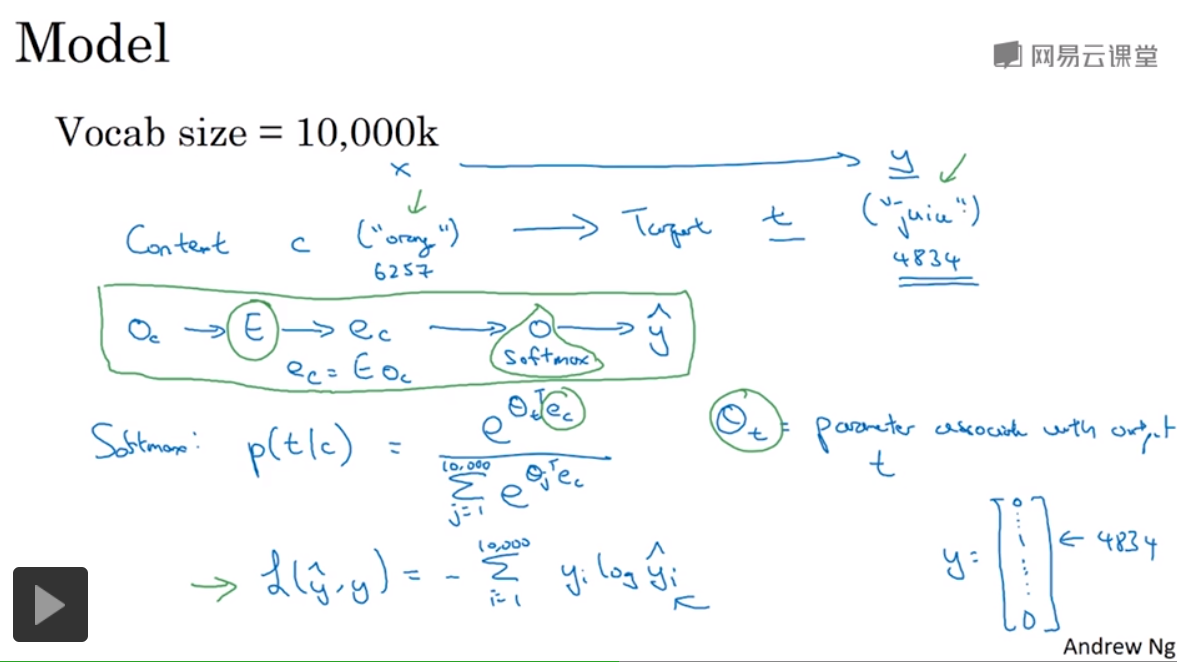
假设10000词的词汇表，有时可能会超过一百万词，要解决的基本监督学习问题本身是学习一种映射关系，从上下文C比如单词orange到某个目标词t，构建一个从输入x到输出y的映射。表示C，先从one-hot，经过嵌入矩阵E得到300维嵌入向量，在神经网络中，将喂入一个softmax单元然后输出，关于模型的细节有：

1. Softmax层预测不同目标词的概率，在给定输入的上下文词为c的情况下， 是一个关于输出目标t的参数

即某个词t和标签相符的概率是多少。

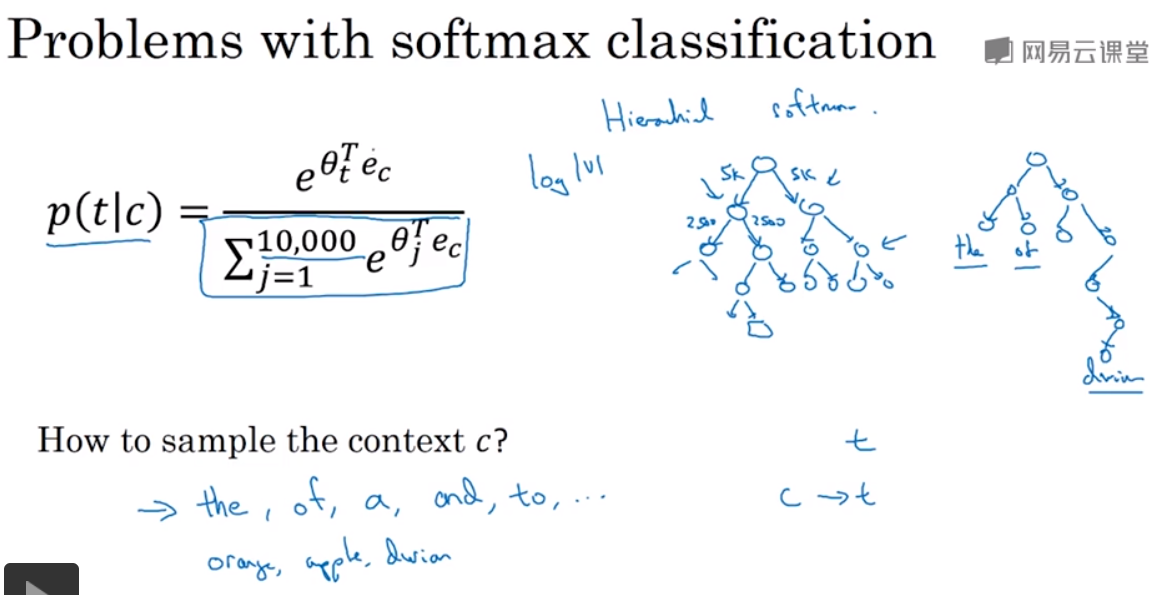
1. 最终softmax的损失函数，

这是目标词y表示为one-hot向量时，常用的softmax损失函数，是计算出来的所有可能目标词的概率。



这是一个学习词嵌入的简化模型和神经网络，其实就是softmax单元，矩阵E有很多的参数，所以矩阵E有对应所有嵌入向量的参数，softmax也有参数，如果优化关于所有这些参数的损失函数，就能得到一个很好的嵌入向量集。

这就是Skip-Gram模型，把一个像orange这样的词作为输入，并预测这个输入词从左数或右数的某个词，即预测输入词的前面或者后面是一些什么词，实际上这个算法会遇到问题：首先问题，计算速度，尤其是softmax模型中，每次计算这个概率值都要进行所有词汇的求和操作，如果词汇更多，求和操作相当慢，扩大词汇表就更加困难。有一些解决方案，加速softmax分类的方法，使用一个分级的softmax分类器，意思就是不是一次确定属于10000类中的哪一类，比如先有个softmax告诉在前5000个还是后面5000个，即二分思想。知道最后找到词准确所在的分类器，这样的树形分类器意味着树上内部的每一个节点都可以是一个二分分类器，比如logistic分类器，所以不再需要一次完成分类对10000个词求和了。实际上这样的分类树计算成本和词汇表大小的对数成正比，而不是词词汇表大小的线性函数。实践中分级softmax数不会使用一颗完全平衡的分类树，或者说左边和右边词数相同的对称树，实际上不常用的词会在树的最深处，常用的在上面，因为常用的词你想要少量检索就能获得。



如何对上下文C进行采样，一种选择是对语料库均匀并且随机的采样，发现the of a and等词出现的相当频繁，并且上下文到目标词的映射会相当频繁得到这些种类的词，但是其他正经词就不会频繁出现。可能不会想要你的训练集都是出现的很频繁的词，这会使你花大部分力气来更新这些频繁出现的词，真正想要的是花时间去出来想orange这些相对少出现的词的嵌入，实际上词P(c)的分布并不是单纯的在训练集语料库上均匀且随机的采样得到的，而是采用了不同的策略来平衡常见词和不常见词。这就是word2vec的Skip-Gram模型。

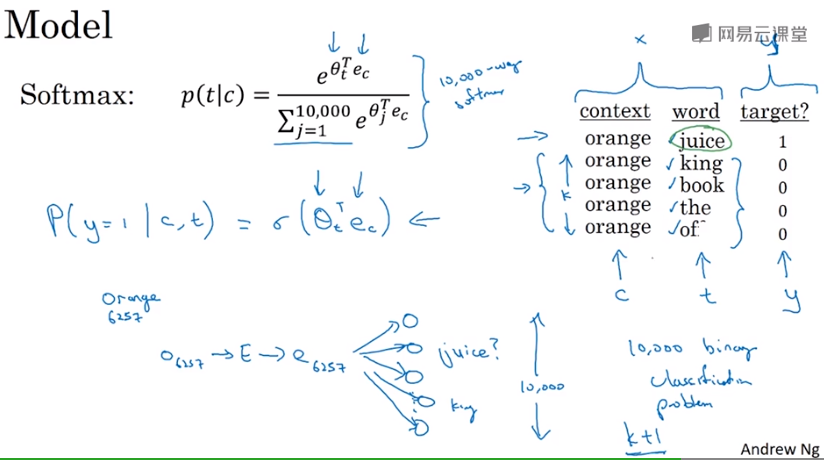
有两个版本的word2vec模型，Skip-Gram是其中一个，和CBOW，连续词袋模型，获得中间词两边的上下文，然后用周围的词去预测中间的词，这个模型也很有效，也有优点和缺点。Skip-Gram的问题依然是softmax这个步骤的计算成本非常昂贵，在分母需要对词汇表中的所有词求和。

## 负采样

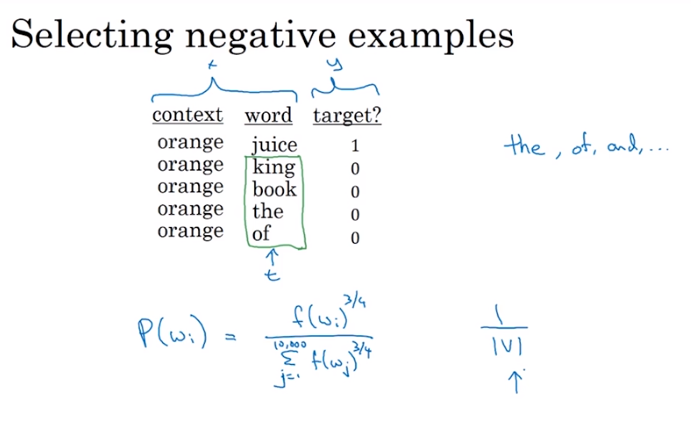
改善过的学习问题负采样，可以做到和Skip-Gram模型相似的事，但是使用了更高效的算法，可以改善它的缺点，softmax计算成本高的问题。构建一个新的监督学习问题，问题是，给定一对单词比如orange和juice，要去预测这是否是一对上下文词和目标词，在上面例句中，这就是个正样本，orange和king就是一个负样本，标为0。所以要做的就是采样得到一个上下文词和一个目标词，这样就生成一个正样本，和在word2vec中的生成方式一样，先采样一个C，再在多少词距里面采样一个t，这就生成表的第一行，再产生一个负样本的方法就是用orange作为c，再在字典中随机选一个词，标为0，再随机选择其他无关的词生成负样本。总结，方式就是选择一个上下文词，在选一个目标词就是一个正样本，给定几次用相同上下文词，再去字典里随机选择k个词组成负样本。如果我们选到的词刚好出现在词距内也无所谓。接下来构造监督学习问题，算法输入x就是这对词，要去预测目标的标签，即y。这个学习就是要分辨不同词组来自不同的采样方式，这就是如何生成训练集的方法，那么如何选择k呢。笑的数据集的话，k为5-20比较好，如果数据集比较大，k就小一点，比如2-5。数据集越小，k越大。在这个例子中，使用k=4。

下面说学习从x到y的映射的监督学习模型，左边是softmax模型，右边分别是c-t-y的训练集，首先构建一个logistic回归模型，给定输入的c和t对的条件下y=1的概率，这个模型基于logistic，但不同的是将一个sigmoid函数用于，参数和之前一样，对每一个可能的目标词有一个参数向量以及可能的上下文词的嵌入向量，这个意思是有10000个参数，哪一个作为选出来的目标词就是，用下面这个公式估计y=1的概率。

如果有k个样本，那么可以看成1：k的正负样本比例，即每个正样本都有k个对应的负样本哎训练一个类似logistic回归的模型，神经网络的结构是首先对于orange这个词有一个one-hot向量，经过嵌入矩阵E得到，然后有10000个可能的logistic回归分类问题，其中一个可能是用来判断目标词是否是juice的分类器，然后还有其他的分类器来判断目标词是否会是其他词，来预测词汇表中这些可能的单词。把这些看成10000个二分类logistic回归分类器，但并不是每次迭代都训练全部10000个，只训练其中的5个，（1+4=5），要训练对应真正目标词的哪一个分类器，再训练4个随机选取的负样本。所以不再使用一个巨大的10000维的softmax层，因为计算成本很高，而是转化为10000个简单的二分类问题，并且每次迭代要做的只是训练其中的几个，k+1个。这就是负采样。在每次迭代中，选择k个不同的随机的负样本对去训练你的算法。



另一个重点就是如何选取负样本，办法就是可以根据其在语料库中的经验频率在选择候选目标词进行采样，通过词出现的频率进行采样，但这会导致在like the of a这些词上有很高的频率，另一个极端是用均匀并且随机地抽取负样本，这对于英文单词的分布是非常没有代表性的。所以根据研究者的经验，这个经验值的效果最好，它位于这两个极端的采样方法之间，既不用经验频率，即观察到的英文文本的分布，也不用均匀分布，做的是对词频的次方除以整体值进行采样，所以是观察到的在语料库中的某个英文单词的词频，通过次方计算，使其处于完全独立的分布和训练集的的观测分布之间。通过转化为logistic回归可以很好的学习词向量。



## Glove词向量

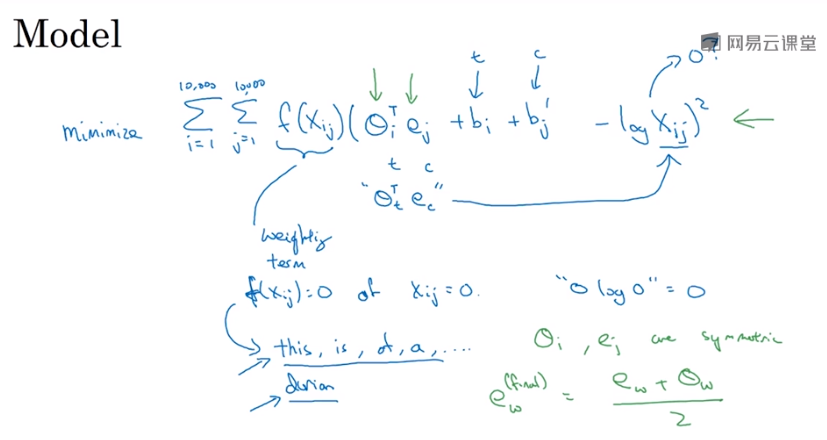
这也是一种学习词嵌入向量的算法，它不如word2vec和Skip-Gram用的多，但因为简单也有人用。Glove表示用词表示的全局变量，global vectors for word representation，之前通过挑选语料库中相近的两个词，举一对词上下文和目标词的例子，Glove做的是使其关系开始明确化，表示次数，单词i在单词j的上下文中出现的次数，这里的i和j就和t和c的功能一样，所以等同于，也可以遍历训练集然后数出单词i和单词j在不同的上下文中出现的个数，单词t和c共有多少次出现在不同语境中。根据上下文词和目标词的定义，，事实上如果将上下文和目标词的范围定义为出现于左右各10个词以内的话，那么就会有一个对称关系，但是如果只是定义在目标前一个词的话，就不会有这种对称关系了。对于Glove算法可以定义上下文和目标词为任意两个位置相近的单词，就是一个可以获取单词i和单词j出现位置相近时或是彼此接近的频率的计数器，Glove做的就是像这样优化，将它们之间的差距进行最小化处理，

后面部分是告诉你这两个单词之间有多少联系，或者说关联程度如何，换句话说它们同时出现的频率是多少，这个是由影响的，然后就是解决参数和e，准备用梯度下降法来最小化求和后的内容：

这样它们的输出能够对这两个单词同时出现的频率进行良好的预测，细节处理：如果，那么log0就是未定义的，负无穷大，所以要对时进行求和，是一个额外的加权项，如果，那么，所以上面这个公式就是只统计了上下文和目标词关系里连续出现至少一次的词的词对的和。的作用就是有些词出现的比较频繁，比如this is等，这些叫做停止词，但是在频繁词和不常用词之间有连续区，当然也还有一些不常用词，还是想将其考虑在内，但又不像常用词那么频繁，因此这个加权因子就可以是一个函数，对不常用的词也能给予大量意义的运算，同时给那些出现的非常频繁的词给不至于过分的权重，因此，就有一些对加权函数f的选择有启发性原则，就是既不给这些词过分的权重，也不会给不常用词太小的权重。

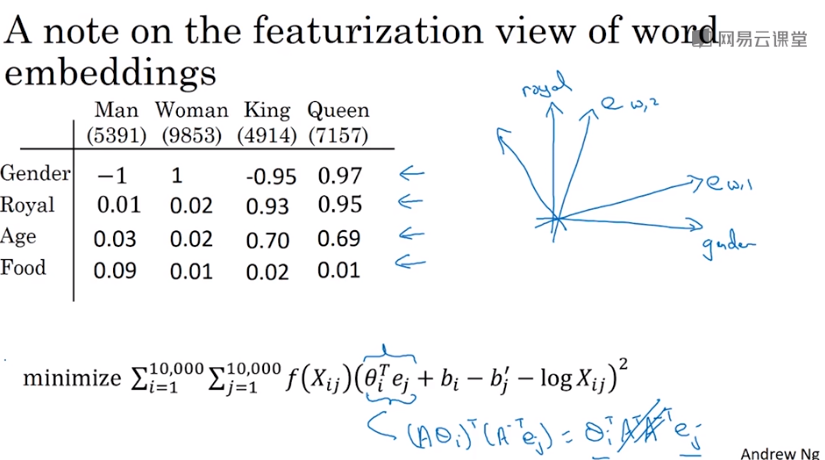
关于和e，现在是完全对称的，所以和也是对称的，从数学函数的角度，功能是相近的，可以将它们点到或者进行排序，实际上都输出了最佳结果，因此一种训练算法的方式是一致地初始化和e，然后使用梯度下降法来最小化输出，当每个词都处理完之后，输出平均值。所以给定一个词w，就有

就是因为这个参数是对称的，而不像之前的模型，有不一样的功能，所以不能像之前一样取平均值。



关于算法的一个疑惑，最小化的公式太简单，仅仅是优化一个这样的二次代价函数，是如何能够学习有意义的词嵌入的呢，但结果证明确实有效。发明这个算法的过程是以历史上最为复杂的模型，以后后来的word2vec等模型为基础，同时希望能简化之前的算法得到的。

之前我们举例子，词嵌入向量的每一行对应一种特性，而在真正的算法中，不能保证嵌入向量的独立组成部分是能够理解的。也就是说算法选择的轴可能不是gender royal等，而是其他方向的，并且可能还不是正交的。所以很难单独的对E解释出每一行到底表示的是什么特性。从线性代数的角度，可以理解。

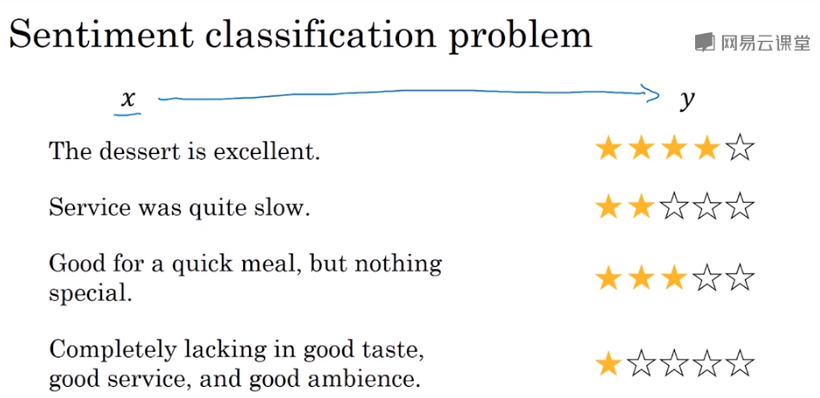


总结学习词嵌入的算法：一般网络，word2vec，负采样，Glove词向量。在2.5节中，举个简单例子想要预测出一句话的orange后面的juice，简单的说了如何使用一个嵌入矩阵E来对输入进行表示，并且经过softmax来预测输出，最后进行梯度下降优化参数。Word2vec中讲到使用上下文和附近的一个目标词，通过计算，预测可能的不同目标词的概率，这种会输出10000个结果，优化损失函数计算太复杂，所以提出了分级softmax的方法。负采样则更好的解决softmax计算量大的问题，通过是采用一个正样本和k个负样本，减少了计算压力，采用词对，判断这是一个正样本还是负样本的方法，将问题简化成了k+1个logistic回归的问题。Glove算法则是通过计算一个简单的代价函数即可完成，样本是一个上下文和目标词组成，然后有对称关系，有权值项可以控制频率次数的权重关系。

## 情感分类

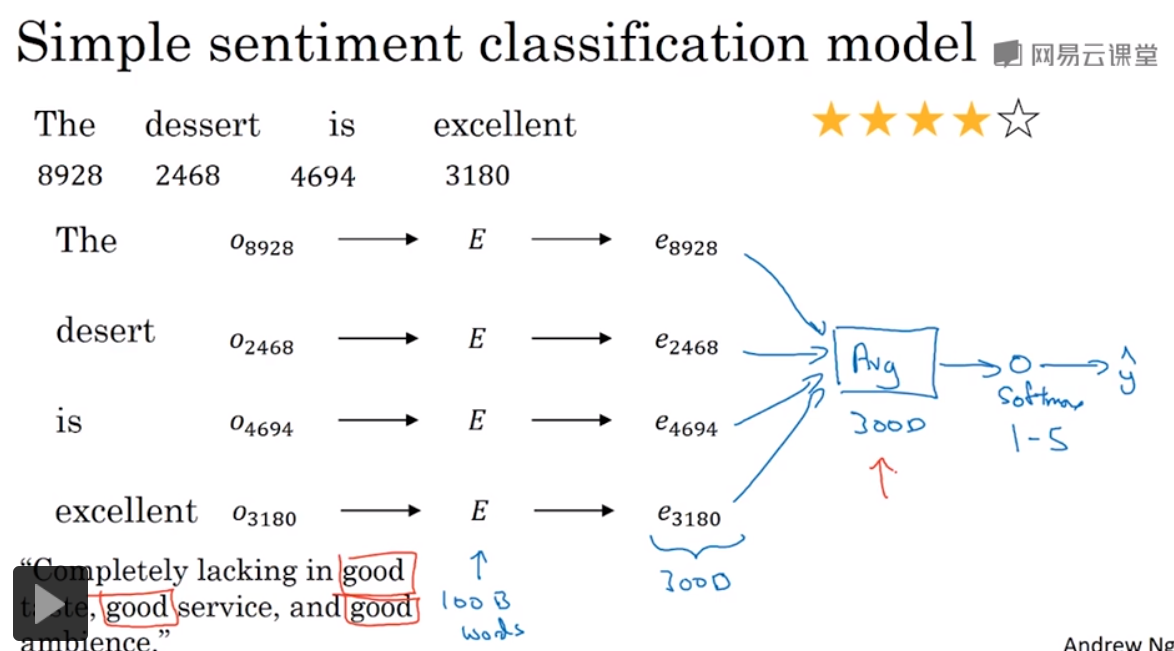
Sentiment classification情感分类任务就是看一段文本，然后分辨这个人对讨论事物的态度。这是NLP种最重要的模块之一，情感分类最大的挑战是可能标记的训练集并没有那么多，但是有了词嵌入，即使只有中等大小的标记的训练集，也能构建一个不错的情感分类器。

输入x是一段文本，输出y是要预测的相应情感，所以需要训练一个从x到y的映射，基于下图中这样标记的数据集，那么就可以收集大家对这个餐馆的评价。如果有一个情感分类器，就可以看一段文本然后分析出这个人对你的餐馆的评论的情感是正面的还是负面的，就可以一直记录餐馆是否存在问题。

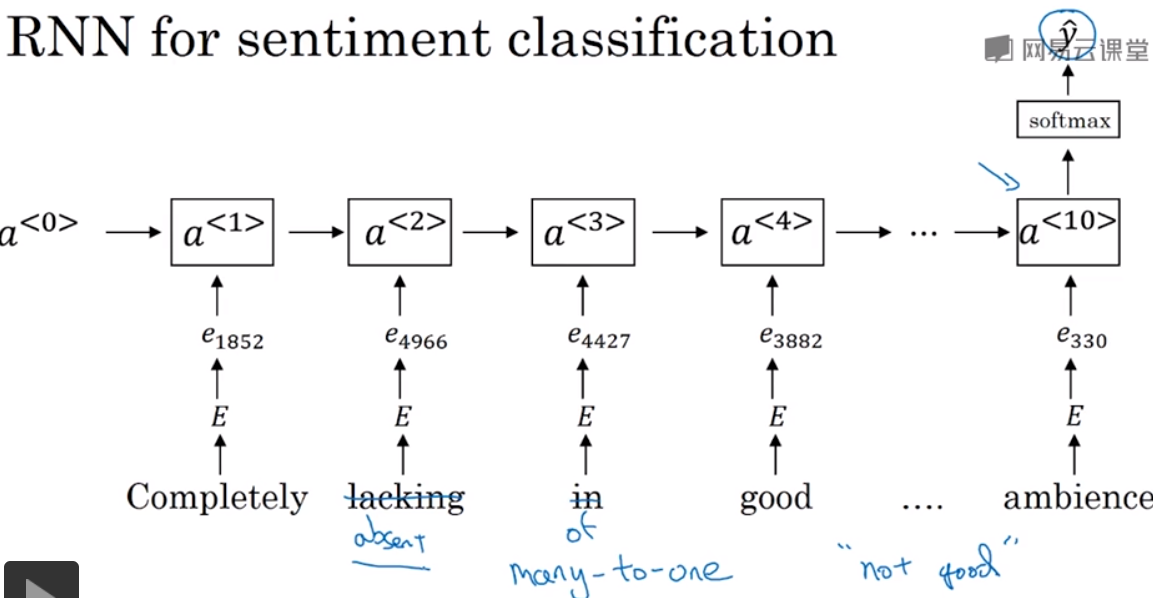


情感分类最大挑战：标记的数据集没有那么多，训练集大大小小从10000到100000个单词都很常见，甚至有时候小于10000个词，采用了词嵌入，能够带来更好的效果，尤其是只有很小的训练集时。这一节会讲几个不同的算法。

下面是一个简单的情感分类的模型，首先对于一段文本，在词典中找这些词，通常用10000个词的词汇表，构建一个分类器可以把这段文本映射成输出4颗星。首先对于The dessert is excellent.找到相应的one-hot表示，乘以矩阵E，E可以从一个很大的文本集里学习到，然后提出到每个词的嵌入向量。如果在很大的训练集上训练E，这样就会获得很多知识，甚至从有些不常用的词中获取然后应用到你的问题上，即使你的标记的数据集里面没有这些词。接下来，使用这些词的嵌入向量来构建一个分类器，比如都是300维的向量，然后将4个300维向量求和或者求平均，得到一个300维的特征向量，将这个向量输入到softmax分类器，输出，输出5个可能结果的概率值，即5个星级评定。这里用的平均值运算单元，这个适用于任何长短的评论，所以这个平均值的运算效果很好，实际上会把所有单词的意思给平均起来，或者都加起来。



这个算法的一个问题是没考虑词序，尤其是图中的那个负面评价，但出现了好多good这个词，如果还是只把所有单词的词嵌入加起来或者平均，最后的特征向量会有很多good的表示，分类器可能就会认为这是个好评。因此有一个更加复杂的模型，不用简单的把所有的词嵌入都加起来，用一个RNN来做情感分类。首先取这条评论，找出每一个词的one-hot表示，并且得到相应的词嵌入表达，再把这个嵌入向量送到RNN中。RNN的工作就是在最后一步计算一个特征向量，用来预测，这是一个多对一的RNN模型。这个模型就考虑了词的顺序，效果更好。这样训练算法，会得到一个很好的情感分类器。由于词嵌入是在一个更大的数据集上训练到的，效果也会更好，更好的泛化一些没有见过的新的单词，即使后来要预测的一段文本中的某个词没有出现在训练集中，但是可能会出现在单词集里。

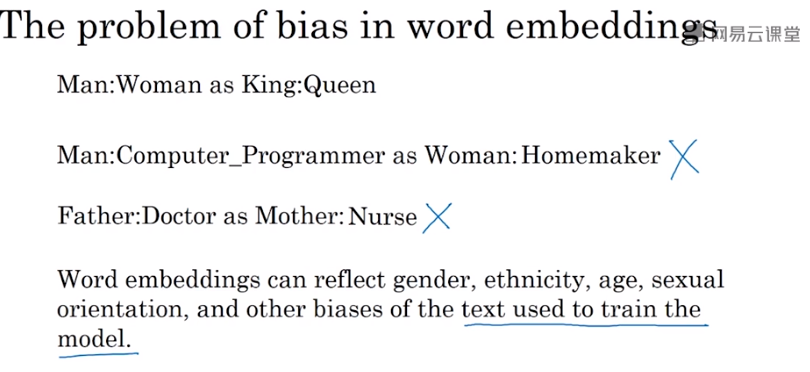


一旦你从网上下载到训练好的嵌入向量集，就能很快的构建一个有效的NLP系统。

## 词嵌入除偏见

深度学习做决策已经越来越普遍，所以应该尽可能的保证它们不受非预期形式偏见影响，比如性别歧视，种族歧视等。这一节将一些关于减少或是消除这些形式的偏见的方法，不是指偏差，而是指性别，种族等方面的偏见。通常内容是词嵌入怎样学习类比像man：woman就像king：queen，不过如果你这样问，man：computer—programmer那么woman：？有研究者发现一个结果，一个已经完成学习的词嵌入可能会输出homemaker，这看起来执行了一个十分不良的性别歧视。如果woman也对应computer—programmer看起来会更合理，并且还有想象，man：doctor那么woman会输出nurse，因此根据训练模型所使用的文本，词嵌入能够反映出性别 种族 年龄 性取向等其他方面的偏见。这些偏见都和社会经济状态相关，因为机器算法已经用来做很重要的决策，所以希望每个人都能有平等的机会，从学业录取，到找工作途径到贷款申请，甚至是判决标准，学习算法都在作出重要决策。所以应该尽量修改学习算法，尽可能减少或者理想化，消除这些非预期类型的偏见是很重要的。

词嵌入能够轻易学会用来训练模型文本中的偏见内容，所以算法获得的偏见内容就能反映出人们写作中的偏见，对于人工智能，也有减少偏见的方法。所以接下来，讨论减少词嵌入偏见问题的方法。



首先要做1. 辨别出我们想要减少或想要消除的特定偏见的趋势，比如性别歧视，在其他形式的偏见也是通用的。如何辨别呢，使用，，等，将这些值求和再求平均，能够解决这种情况下这个趋势看起来就是性别趋势。而其他方向上的偏差与性别无关，是一个无偏见趋势。这个方向上如果与某个特征轴平行，也可以不平行，因为前面说过某一个特征可能由好几个维度的数据来共同管理。并且这里趋势判断不是简单的求平均，而是使用更加复杂的算法叫做SVU，奇值分解，singular value decomposition。如果对主成分分析很熟悉，奇值分解这个算法的一些方法和主成分很类似。2. 中和步，neutralize：对于那些定义不确切的词，可以将其处理一下避免偏见。有些词本身上就和性别有关，定义中本身就含性别内容，但有些词babysitter doctor就应该是中立的词。也就是说，不含明确性别定义的词，可以将它们在非偏见轴上进行处理，来减少或者消除它们的性别偏差成分。也就是减它们在性别偏差这个轴上的距离，比如投影到性别偏差轴上在一个位置，距离为0。3. 均衡步，equalize pairs，意思是说可能会有这样的词对，grandmother和grandfather，以及boy和girl，只希望性别是其区别。做法就是虽然在性别轴上映射位置不同，但是在其他偏见轴上位置距离相同，或者说投影到的位置一样，这样就可以保证像doctor这些应该中立的词距离男女性别也会一样。最后一点，如何决定哪个词是中立的呢。论文作者做的就是训练一个分类器来尝试解决哪些词是有明确定义的，哪些措施性别确定的，结果表明英语里的很多次在性别上面都是没有明确定义的，只有小部分是跟性别明确相关的。一个线性分类器能够告诉你哪些词能够通过中和步来预见这个偏差趋势，并且在非偏见轴进行处理。需要平衡的词的对数，实际上是很小的。详细的内容可以自己查阅论文。

