1 自然语言处理与词嵌入

自然语言处理（Natural Language Processing，NLP) 是人工智能和语言学领域的学科分支，它研究实现人与计算机之间使用自然语言进行有效通信的各种理论和方法。

1.1 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**
- **词嵌入**

1.2 词嵌入

前面介绍过，处理文本序列时，通常用建立字典后以 one-hot 的形式表示某个词，进而表示某个句子的方法。这种表示方法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，从而孤立了每个词，无法表现各个词之间的相关性，满足不了 NLP 的要求。

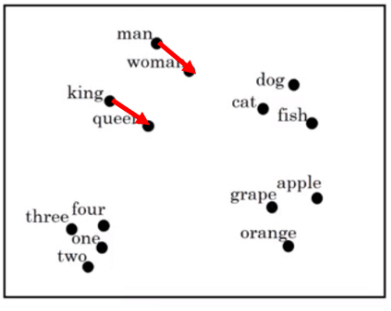
换用特征化表示方法能够解决这一问题。我们可以通过用语义特征作为维度来表示一个词，因此语义相近的词，其词向量也相近。

词嵌入（Word Embedding） 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。

	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

如上图中，各列分别组成的向量是词嵌入后获得的第一行中几个词的词向量的一部分。这些向量中的值，可代表该词与第一列中几个词的相关程度。

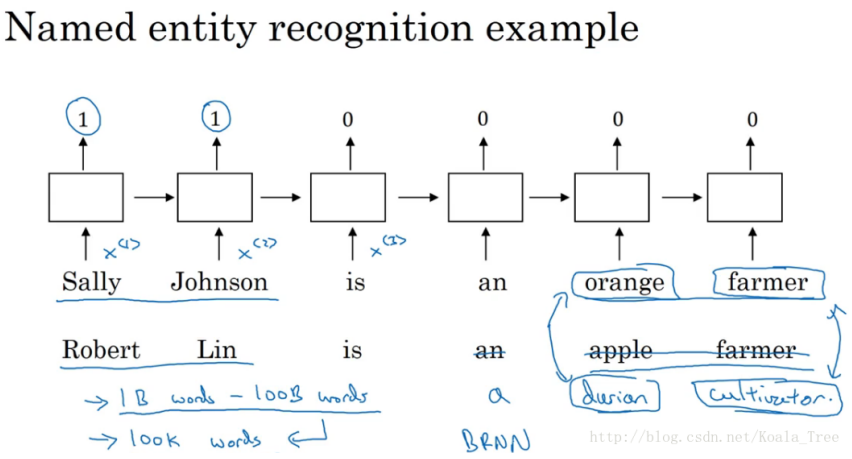
使用 2008 年 van der Maaten 和 Hinton 在论文 Visualizing Data using t-SNE (https://www.seas.harvard.edu/courses/cs281/papers/tsne.pdf) 中提出的 t-SNE 数据可视化算法，将词嵌入后获得的一些词向量进行非线性降维，可到下面的映射结果：



其中可发现，各词根据它们的语义及相关程度，分别汇聚在了一起。

对大量词汇进行词嵌入后获得的词向量，可用来完成**命名实体识别（Named Entity Recognition）**等任务。

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。



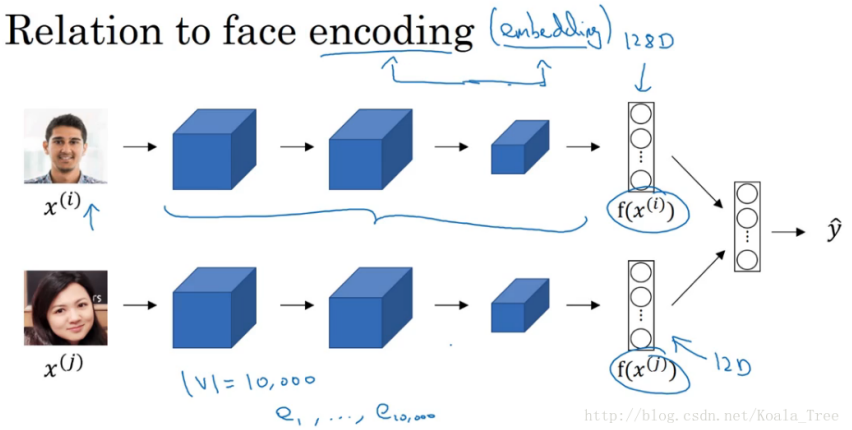
词嵌入和迁移学习

用词嵌入做迁移学习可以降低学习成本，提高效率。其步骤如下：

- 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；
- 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；
- 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。

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，未学习过的词无法识别。

词嵌入和类比推理

好比前面讲过的用 Siamese 网络进行人脸识别过程，使用词嵌入方法获得的词向量可实现词汇的类比及相似度度量。例如给定对应关系 “男性（Man）” 对 “女性（Woman）”，要求机器类比出 “国王（King）” 对应的词汇，通过上面的表格，可发现词向量存在数学关系 “Man - Woman ≈ King - Queen”，也可以从可

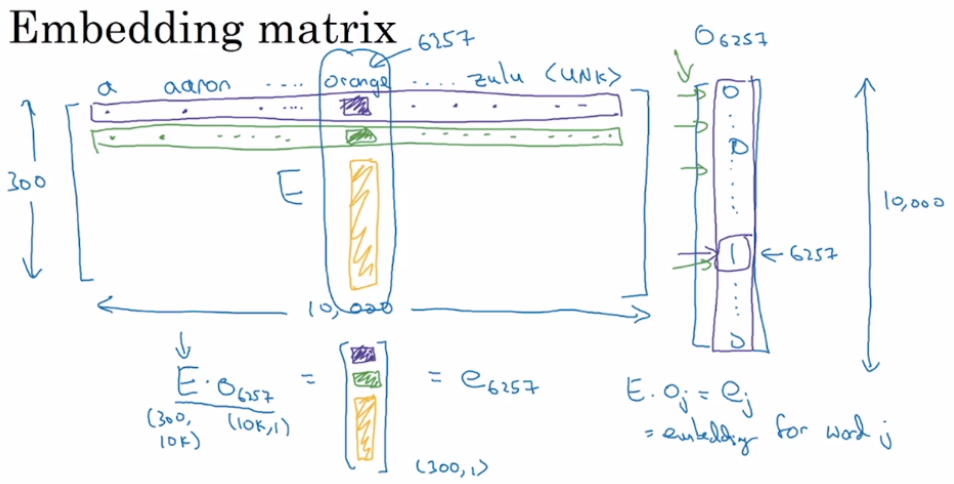
视化结果中看出 “男性（Man）” 到 “女性（女性）” 的向量与 “国王（King）” 到 “王后（Queen）” 的向量相似。则可以有 $e_{man} - e_{woman} \approx e_{king} - e_?$，之后的目标就是找到词向量 w，来找到使相似度 $sim(e_w, e_{king} - e_{man} + e_{woman})$ 最大。

一个最常用的相似度计算函数是余弦相似度（cosine similarity）。公式为：

$$sim(u, v) = \frac{u^T v}{||u||_2 ||v||_2} \tag{1}$$

词嵌入具有的这种特性，在 2013 年 Mikolov 等发表的论文 Linguistic Regularities in Continuous Space Word Representations (https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/rvecs.pdf) 中提出，成为词嵌入领域具有显著影响力的研究成果。

1.3 嵌入矩阵



不同的词嵌入方法能够用不同的方式学习到一个**嵌入矩阵（Embedding Matrix）** E。将字典中位置为 i 的词的 one-hot 向量表示为 o_i，词嵌入后生成的词向量用 e_i 表示，则有：

$$E \cdot o_i = e_i \tag{2}$$

例如字典中包含 10000 个词，每个词的 one-hot 形式就是个大小为 10000×1 的列向量，采用某种方法学习到的嵌入矩阵大小为 300×10000 的话，将生成大小为 300×1 的词向量。

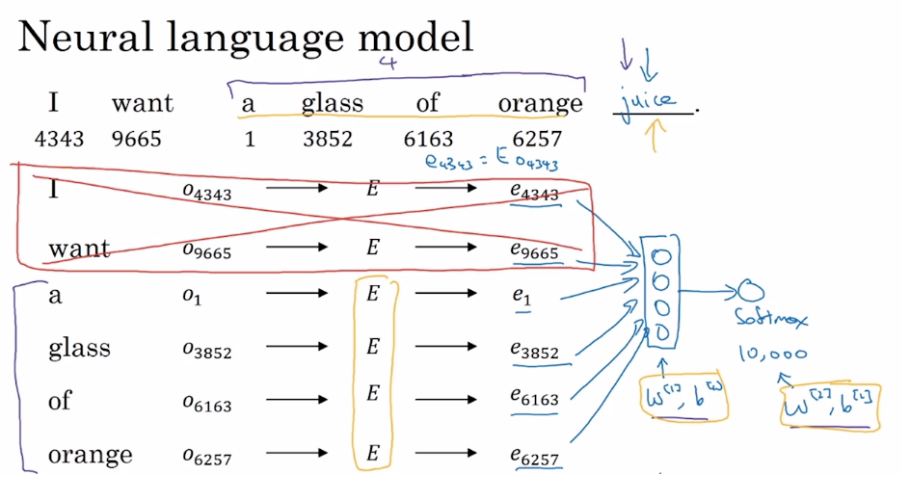
但在实际情况下一般不这么做。因为 one-hot 向量维度很高，且几乎所有元素都是 0，这样做的效率太低。因此，实践中直接用专门的函数查找矩阵 E 的特定列。

1.4 词嵌入方法

1.4.1 神经概率语言模型

采用神经网络建立语言模型是学习词嵌入的有效方法之一。2003 年 Bengio 等人的经典之作 A Neural Probabilistic Language Model (http://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf) 中，提出的神经概率语言模型，是早期最成功的词嵌入方法之一。

模型中，构建了一个能够通过上下文来预测未知词的神经网络，在训练这个语言模型的同时学习词嵌入。例如将下图中上面的句子作为下面的神经网络的输入：



经过隐藏层后，最后经 Softmax 将输出预测结果。其中的嵌入矩阵 E 与 w、b 一样，是该网络中的参数，需通过训练得到。训练过程中取语料库中的某些词作为目标词，以目标词的部分上下文作为输入，训练网络输出的预测结果为目标词。得到了嵌入矩阵，就能通过前面所述的数学关系式求得词嵌入后的词向量。

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）。

1.4.2 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 (SG)：根据词预测目标上下文
- Continuous Bag of Words (CBOW)：根据上下文预测目标词

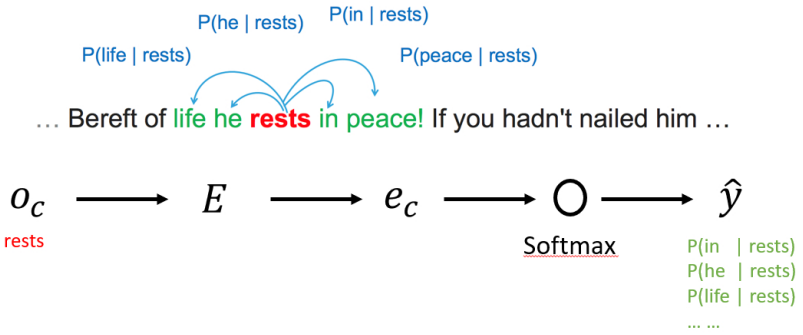
每种语言模型又包含**负采样（Negative Sampling）**和**分级的 Softmax（Hierarchical Softmax）**两种训练方法。

训练神经网络时候的隐藏层参数即是学习到的词嵌入。

Word2Vec（Word To Vectors）是现在最常用、最流行的词嵌入算法，它在 2013 年由 Mikolov 等人在论文 Efficient Estimation of Word Representations in Vector Space (https://arxiv.org/pdf/1301.3781.pdf) 中提出。

1.4.2.1 Skip-Gram 模型

Word2Vec 中的 Skip-Gram 模型，所做的是在语料库中选定某个词（Context），随后在该词的正负 10 个词距（一般情况）内取一些目标词（Target）与之配对，构造一个用 Context 预测输出为 Target 的监督学习问题，训练一个如下图结构的网络：



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：

$$Softmax: p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}} \tag{3}$$

  其中 θ_t 是与输出 t 有关的参数；
- 损失函数：$L(\hat{y}, y) = -\sum_{i=1}^{10,000} y_i \log \hat{y}_i$，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

- 在 Softmax 单元中，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。
- 在实践中，一般采用霍夫曼树（Huffman Tree）而非平衡二叉树，常用词在顶部。

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。

1.4.2.2 CBOW 模型

CBOW 模型的工作方式与 Skip-gram 相反，通过采样上下文中的词来预测中间的词。

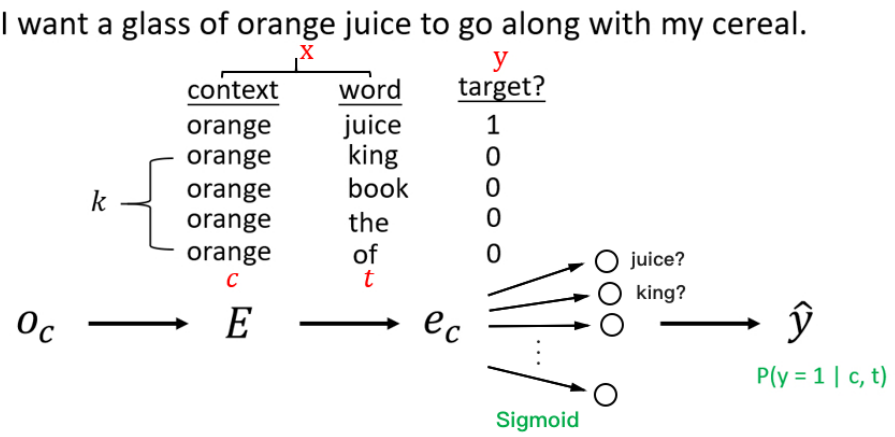
吴恩达老师没有深入去讲 CBOW。想要更深入了解的话，推荐资料：

- 秒懂词向量Word2vec的本质（中文，简明原理）(https://zhuanlan.zhihu.com/p/26306795)
- word2vec 原理推导与代码分析-码农场（中文，深入推导）(http://www.hankcs.com/nlp/word2vec.html)
- 课程 cs224n 的 notes1（英文）(https://github.com/stanfordnlp/cs224n-winter17-notes/blob/master/notes1.pdf)

1.4.2.3 负采样

为了解决 Softmax 计算较慢的问题，Word2Vec 的作者在后续论文 Distributed Representations of Words and Phrases and their Compositionality (https://arxiv.org/pdf/1310.4546.pdf) 中提出了**负采样（Negative Sampling）**模型，进一步改进和简化了词嵌入方法。

负采样模型中构造了一个预测给定的单词是否为一对 Context-Target 的新监督学习问题，采用的网络结构和前面类似：



计算过程

- 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。

改用多个 Sigmoid 输出上下文-目标词（c, t）为正样本的概率：

$$P(y=1|c,t)=\sigma(\theta_t^T e_c) \tag{4}$$

其中，θ_t、e_c 分别代表目标词和上下文的词向量。

之前训练中每次要更新 n 维的多分类 Softmax 单元（n 为词典中词的数量）。现在每次只需要更新 k+1 维的二分类 Sigmoid 单元，计算量大大降低。

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）：

$$p(w_i)=\frac{f(w_i)^{\frac{3}{4}}}{\sum_{j=0}^{m} f(w_j)^{\frac{3}{4}}} \tag{5}$$

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，$f(w_i)$ 是语料库中观察到的某个词的词频。

1.4.3 Glove

GloVe（Global Vectors）是另一种现在流行的词嵌入算法，它在 2014 年由 Pennington 等人在论文 GloVe: Global Vectors for Word Representation (https://nlp.stanford.edu/pubs/glove.pdf) 中提出。

Glove 模型中，首先基于语料库统计了词的共现矩阵 X，X 中的元素为 X_{ij}，表示整个语料库中单词 i 和单词 j 彼此接近的频率，也就是它们共同出现在一个窗口中的次数。之后要做的，就是优化以下代价函数：

$$J = \sum_{i,j}^{N} f(X_{i,j})(\theta_i^T e_j + b_i + b_j - log(X_{i,j}))^2 \tag{6}$$

其中 θ_i、e_j 分别是单词 i 和单词 j 的词向量，b_i、b_j 是两个偏差项，$f()$ 是一个用以防止 $X_{ij} = 0$ 时 $log(X_{i,j})$ 无解的权重函数，词汇表的大小为 N。

根据上下文和目标词的定义，我们发现 X_{ij} 与 X_{ji} 其实是一样的。在这种情况下，即 θ_i 和 e_j 是完全对称的。因此，在训练时可以一致地初始化二者，使用梯度下降法处理完以后取平均值作为二者共同的值。

优化函数的推导过程见：“理解GloVe模型” (https://blog.csdn.net/codertc/article/details/73864097)

最后要说明的是，使用各种词嵌入方法学习到的词向量，并不像最开始介绍词嵌入时展示的表格中 Man、Woman、King、Queen 的词向量那样，其中的值能够代表着与 Gender、Royal 等词的的相关程度，实际上它们大都超出了人们的能够理解范围，难以从某个值中看出与语义的相关程度。

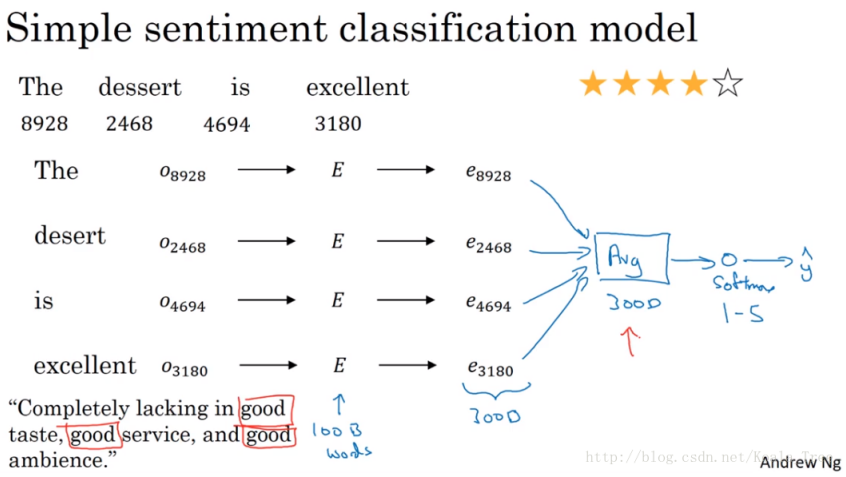
1.5 情感分类（词嵌入应用）

NLP 中的情感分类，是对某段文字中所表达的情感做出分类，实际应用包括舆情分析、民意调查、产品意见调查等等。

训练情感分类模型时，面临的挑战之一可能是标记好的训练数据不够多。然而有了词嵌入得到的词向量，只需要中等数量的标记好的训练数据，就能构建出一个表现出色的情感分类器。

1.5.1 平均值或和预测

- 获取一个训练好的词嵌入矩阵 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 $\hat{y}$

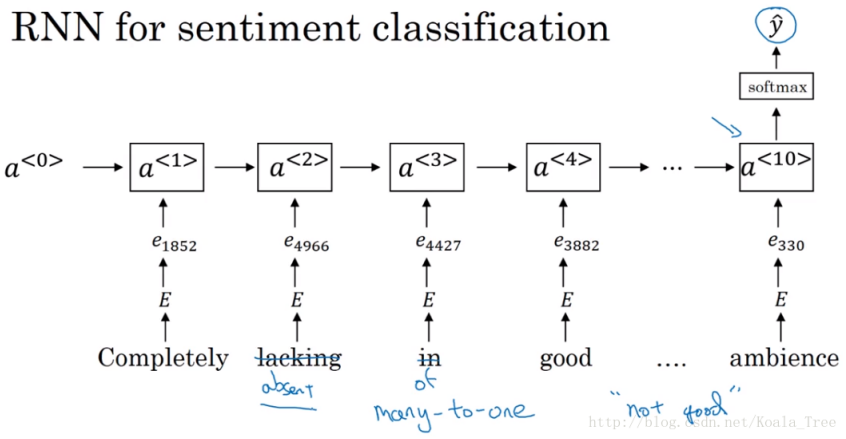


优点是适用于任何长度的文本；缺点是没有考虑词的顺序，对于包含了多个正面评价词的负面评价，很容易预测到错误结果。

1.5.2 RNN 模型

采用RNN能实现一个表现更加出色的情感分类器，此时构建的模型如下：

- 获取一个训练好的词嵌入矩阵 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 $\hat{y}$。



优点：考虑了词序，效果好很多。

这是一个“多对一”结构的循环神经网络，每个词的词向量作为网络的输入，由Softmax输出结果。由于词向量是从一个大型的语料库中获得的，这种方法将保证了词的顺序的同时能够对一些词作出泛化。

1.6 词嵌入除偏

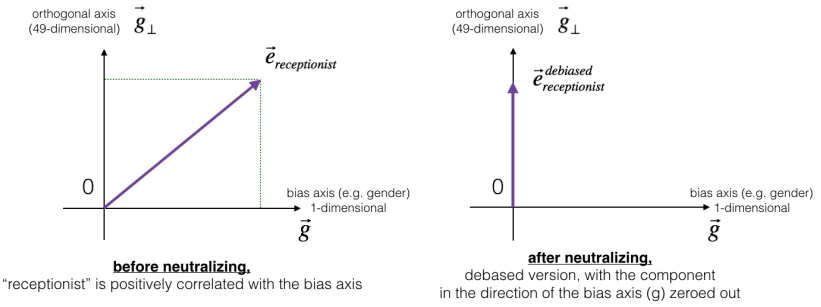
在词嵌入过程中所使用的语料库中，往往会存在一些性别、种族、年龄、性取向等方面的偏见，从而导致获得的词向量中也包含这些偏见。比如使用未除偏的词嵌入结果进行词汇类比时，“男性（Man）”对“程序员（Computer Programmer）”将得到类似“女性（Woman）”对“家务料理人（Homemaker）”的性别偏见结果。2016 年 Bolukbasi 等人在论文 Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings (https://arxiv.org/pdf/1607.06520.pdf) 中提出了一些消除词嵌入中的偏见的方法。

这里列举消除词向量存在的性别偏见的过程，来说明这些方法。（摘自第二周课后作业）

中和本身与性别无关词汇

中和（Neutralize）“医生（doctor）”、“老师（teacher）”、“接待员（receptionist）”等本身与性别无关词汇中的偏见，首先计算 $g = e_{woman} - e_{man}$，用“女性（woman）”的词向量减去“男性（man）”的词向量，得到的向量 g 就代表了“性别（gender）”。假设现有的词向量维数为 50，那么对某个词向量，将 50

维空间分成两个部分：与性别相关的方向 g 和与 g **正交**的其他 49 个维度 $g_{\perp}$。如下左图：



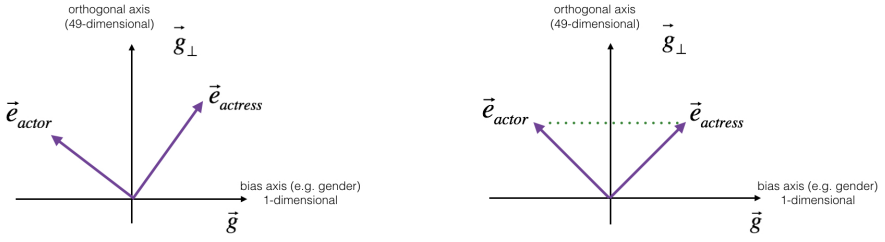
除偏的步骤，是将要除偏的词向量，左图中的 $e_{receptionist}$，在向量 g 方向上的值置为 0，变成右图所示的 $e^{debiased}_{receptionist}$ 所用的公式如下：

$$e^{bias}_{component} = \frac{e \cdot g}{||g||_2^2} \times g \tag{7}$$

$$e^{debiased}_{receptionist} = e - e^{bias}_{component}$$

均衡本身与性别有关词汇

对 "男演员（actor）"、"女演员（actress）"、"爷爷（grandfather）" 等本身与性别有关词汇，如下左图，假设 "女演员（actress）" 的词向量比 "男演员（actor）" 更靠近于 "婴儿看护人（babysit）"。中和 "婴儿看护人（babysit）" 中存在的性别偏见后，还是无法保证它到 "女演员（actress）" 与到 "男演员（actor）" 的距离相等。对一对这样的词，除偏的过程是**均衡（Equalization）**它们的性别属性。



均衡过程的核心思想是确保一对词（actor 和 actress）到 $g_{\perp}$ 的距离相等的同时，也确保了它们到除偏后的某个词（babysit）的距离相等，如上右图。

对需要除偏的一对词 w_1、w_2，选定与它们相关的某个未中和偏见的单词 B 之后，均衡偏见的过程如下公式：

$$\mu = \frac{e_{w1} + e_{w2}}{2} \tag{8}$$

$$\mu_B = \frac{\mu \cdot \text{bias\_axis}}{||\text{bias\_axis}||_2^2} * \text{bias\_axis}$$

$$\mu_\perp = \mu - \mu_B$$

$$e_{w1B} = \frac{e_{w1} \cdot \text{bias\_axis}}{||\text{bias\_axis}||_2^2} * \text{bias\_axis}$$

$$e_{w2B} = \frac{e_{w2} \cdot \text{bias\_axis}}{||\text{bias\_axis}||_2^2} * \text{bias\_axis}$$

$$e_{w1B}^{corrected} = \sqrt{|1 - ||\mu_\perp||_2^2|} * \frac{e_{\text{w1B}} - \mu_B}{|(e_{w1} - \mu_\perp) - \mu_B)|}$$

$$e_{w2B}^{corrected} = \sqrt{|1 - ||\mu_\perp||_2^2|} * \frac{e_{\text{w2B}} - \mu_B}{|(e_{w2} - \mu_\perp) - \mu_B)|}$$

$$e_1 = e_{w1B}^{corrected} + \mu_\perp$$

$$e_2 = e_{w2B}^{corrected} + \mu_\perp$$