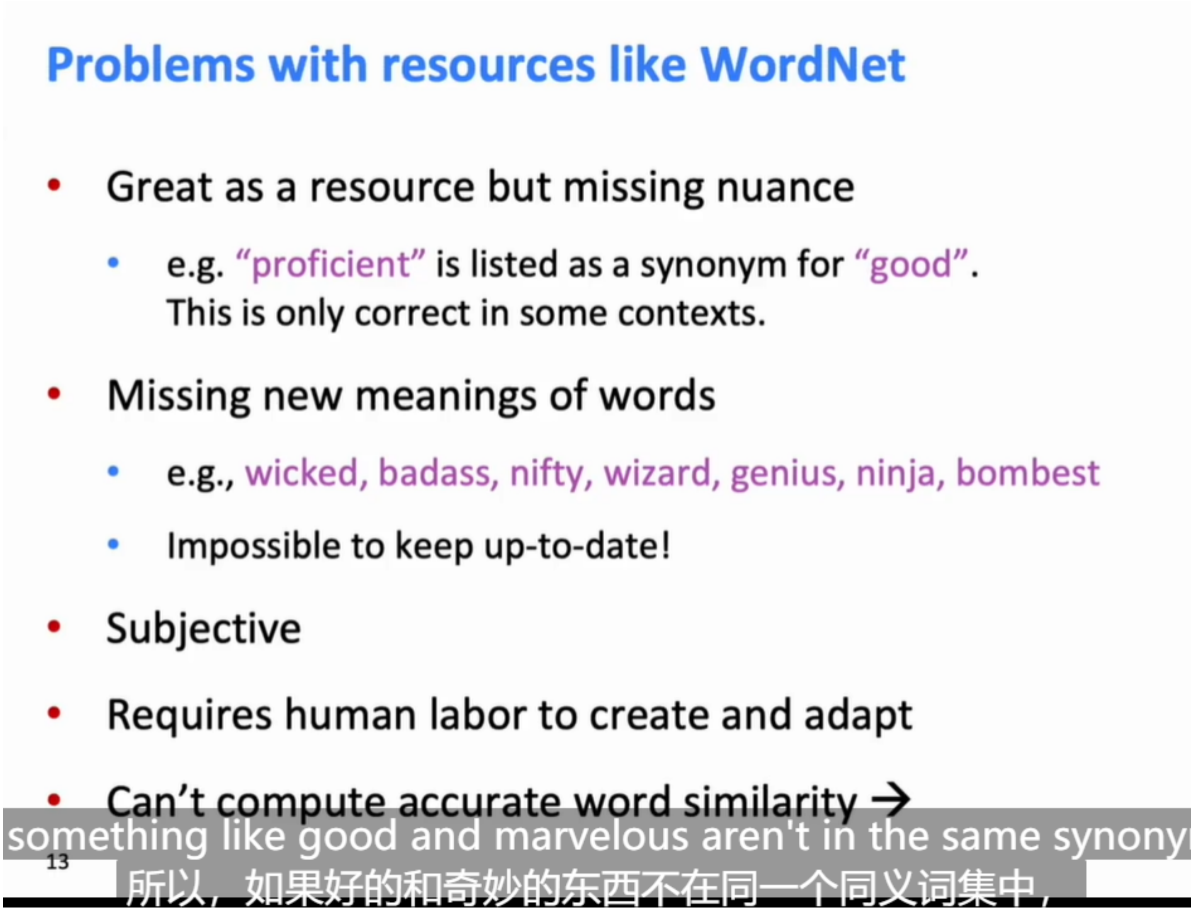
Word2Vec

WordNet

nlp的瑞士军刀，可以当一个nlp工具库来用

缺点



传统NLP的Word表示

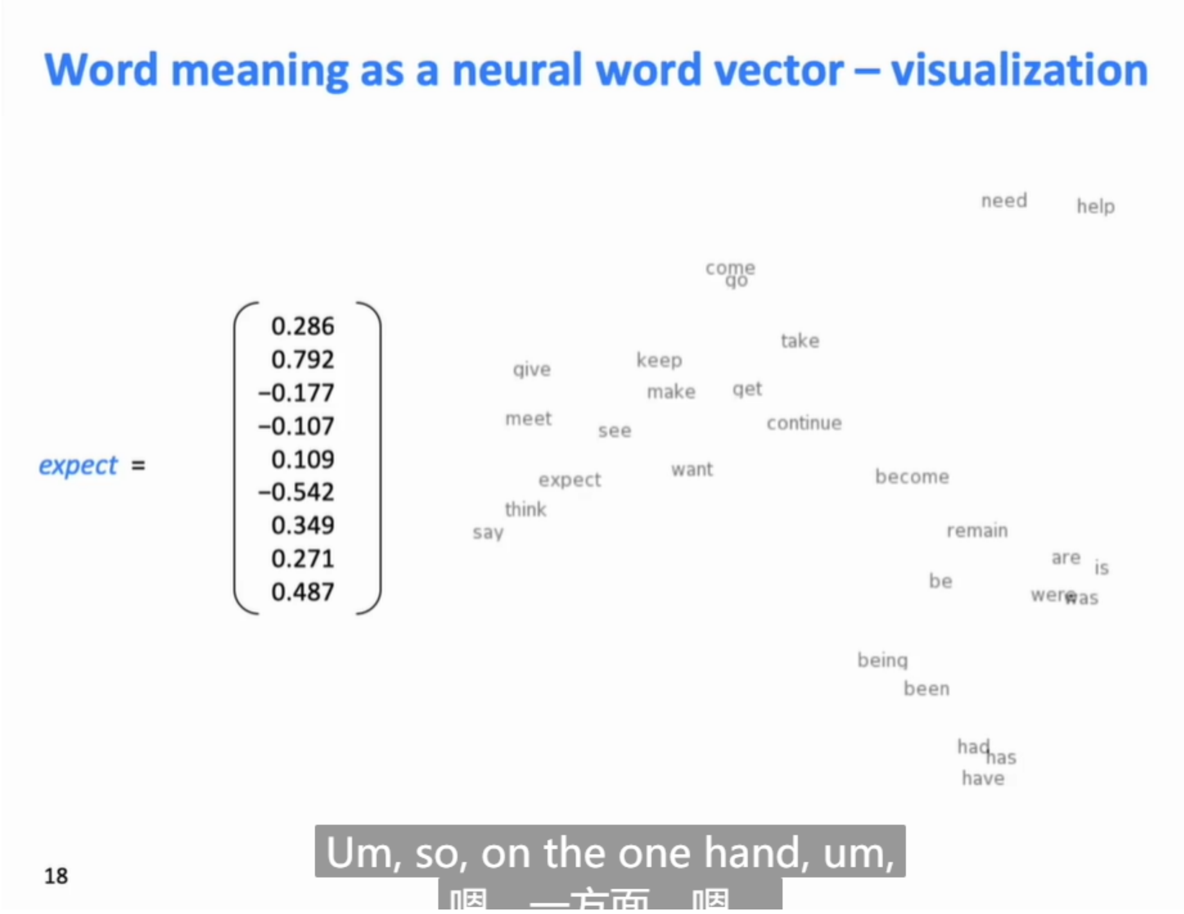
用One-hot编码表示单词，但是有两个问题：

- 单词数量太多了，向量需要上百万维。
- One-hot都是正交的， 无法得到word之间的相似性。

Distributed Representation

"如果你知道什么时候该用一个word，什么时候不该用一个word，那么你就已经理解了它的meaning。"

word vector就是一种distributed representation，即一个多维向量表示一个word。下图是100维投影到2维的可视化图像。



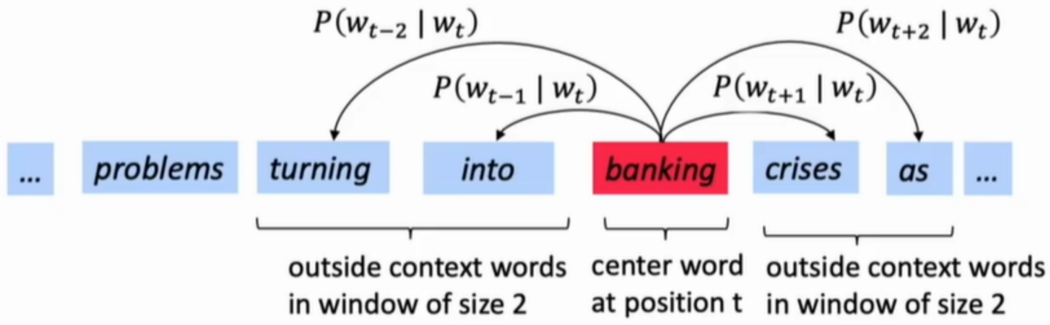
Word2vec算法

要利用word上下文的信息，word在中心称为center，word作为上下文称为context。

用u和v两个向量分别表示一个word在context和在center的向量。

Word2Vec Overview

- Example windows and process for computing $P(w_{t+j} \mid w_t)$



Word2vec: objective function

For each position $t = 1, \dots, T$, predict context words within a window of fixed size m, given center word w_j.

$$\text{Likelihood} = L(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} P(w_{t+j} \mid w_t; \theta)$$

θ is all variables to be optimized

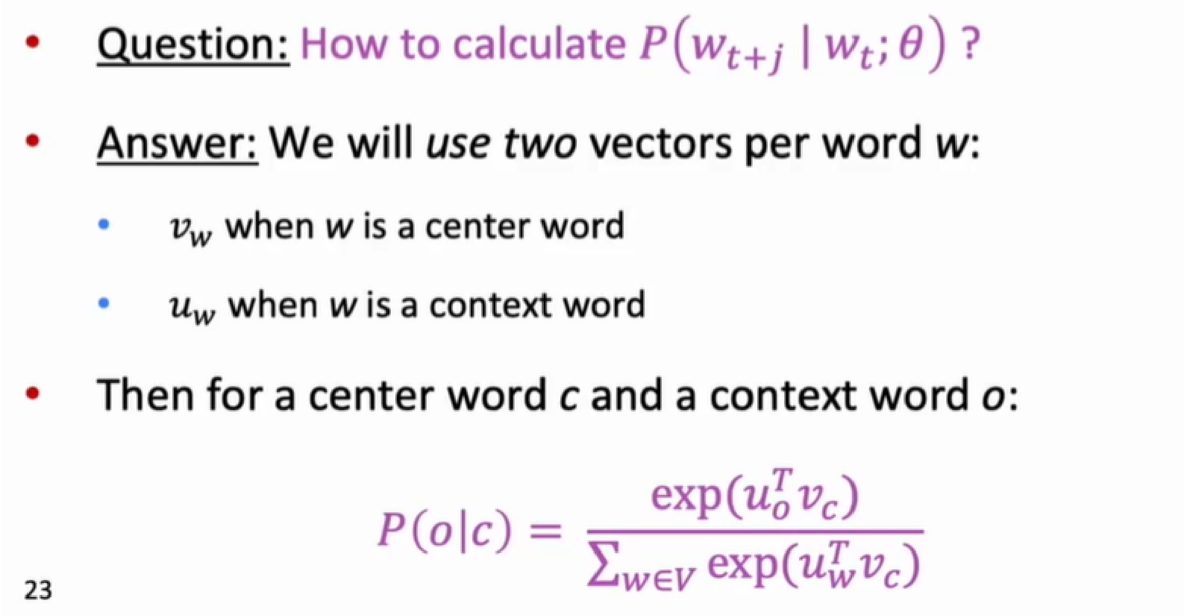
sometimes called *cost* or *loss* function

The objective function $J(\theta)$ is the (average) negative log likelihood:

$$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \ne 0}} \log P(w_{t+j} \mid w_t; \theta)$$

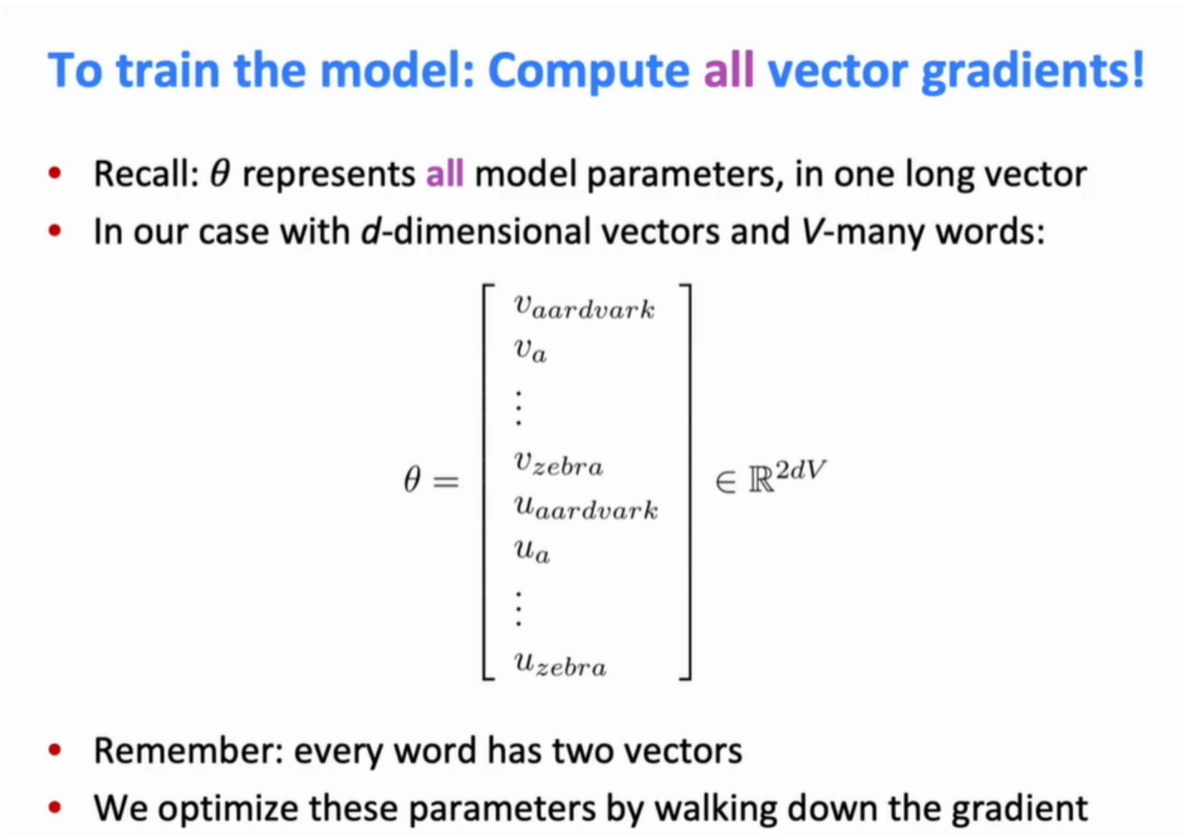
Minimizing objective function ⟺ Maximizing predictive accuracy

上图是最大似然估计(MLE)，取负取log之后变成要minimize的目标函数。



上图是如何用词向量计算center出现时，context出现的条件概率。

方法是用softmax，其中V是词典里所有单词。想象某个center和context关联性极强，那么P(o|c)趋近于1。



上图是最简单的模型下的θ参数，即词典里每个wordvec的每个维度都视作一个参数。

手推求导

推导得到目标函数$J(\theta)$的导数。

$$\max \quad J'(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m \\ j \ne 0}} p(w_{t+j} \mid w_t ; \theta)$$

$$\text{Minimize } J(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m \\ j \ne 0}} \log p(w_{t+j} \mid w_t)$$

change!

$$p(o \mid c) = \frac{\exp(u_o^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)}$$

$$\frac{\partial}{\partial v_c} \overset{f}{\log} \underbrace{\sum_{w=1}^{V} \exp(u_o^T v_c)}_{z(v_c)}$$

$$\frac{1}{\sum_{w=1}^{V} \exp(u_o^T v_c)} \cdot \sum_{x=1}^{V} \frac{\partial}{\partial v_c} \underset{f}{\exp} \underbrace{(u_x^T v_c)}_{z(v_c)}$$

$$\sum_{x=1}^{V} \exp(u_x^T v_c) \cdot \frac{\partial}{\partial v_c} u_x^T v_c$$

$$\sum_{x=1}^{V} \exp(u_x^T v_c) \cdot u_x$$

$$\frac{\partial}{\partial v_c} \log p(o \mid c) = u_o - \sum_{x=1}^{V} \underbrace{\boxed{\frac{\exp(u_x^T v_c)}{\sum_{w=1}^{V} \exp(u_w^T v_c)}}}_{p(x \mid c)} \cdot u_x$$

$$= u_o - \sum_{x=1}^{V} p(x \mid c) \cdot u_x$$

导数$u_o - \sum_{x \in V} p(x|c) u_x$的直观意义是actual context word和expected context word(各个context的加权和)的差值。这样最后所有u_i都变成$\sum p(x|c) u_x$吗？不是，这里是对v_c求导。v_c最后期望达到actual context word = expected context word

下图证明了为什么$\frac{\partial(u^T v)}{\partial v} = u$。