word2vec模型

谢海华 助理研究员

北京雁栖湖应用数学研究院（BIMSA）

大数据与人工智能（BDAI）

2022.03

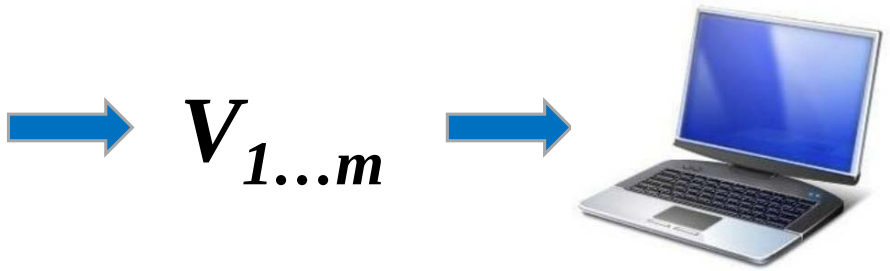
背景及问题

文本及词语表示

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

Former US Secretary of State Colin Powell died on Monday of Covid-19 complications. His family announced that he was fully vaccinated. He was 84 years old, and had multiple myeloma, a blood cancer.

Health officials worry that anti-vaccine activists will seize upon Powell's death to make the claim that vaccines don't work. If you can still die after being vaccinated for Covid-19, what's the point of getting the vaccine?



- **词的表示（Word Representations / Word Embeddings）**
  - 表示方法：One-hot Representation, Distributed Representation
  - 捕获语言特征（linguistic features），包括：语义特征（semantic features，词与句子的语义），句法特征（syntactic features，语句的结构组织以及句中词语之间的依赖关系）。
  - 反映语言现象：一词多义（polysemy），指代（anaphora），语用学（pragmatics，现多指言外之意）等。

文本及词语表示

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

One-hot Representation：

I	0, …, 0, 0, 1
Me	0, …, 0, 1, 0
You	0, …, 1, 0, 0

Distributed Representation：

I	0.03892, …, 0.05738, 0.02657, 0.07856
Me	0.03618, …, 0.05261, 0.03272, 0.07729
You	0.02973, …, 0.05421, 0.03309, 0.07482

word2vec

```
In [8]: model.similarity("woman", "man")
Out[8]: 0.71207063453821218

In [10]: model.doesnt_match("breakfast cereal dinner lunch".
Out[10]: 'cereal'

In [11]: model.similarity("woman", "girl")
Out[11]: 0.67365416785207421
```

```
In [10]: result = model.most_similar(u"青蛙")

In [11]: for e in result:
    print e[0], e[1]
   ....:
老鼠 0.559612870216
乌龟 0.489831030369
蜥蜴 0.478990525007
猫 0.46728849411
鳄鱼 0.461885392666
蟾蜍 0.448014199734
猴子 0.436584025621
白雪公主 0.434905380011
蚯蚓 0.433413207531
螃蟹 0.4314712286
```

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

基础知识

Sigmoid函数

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

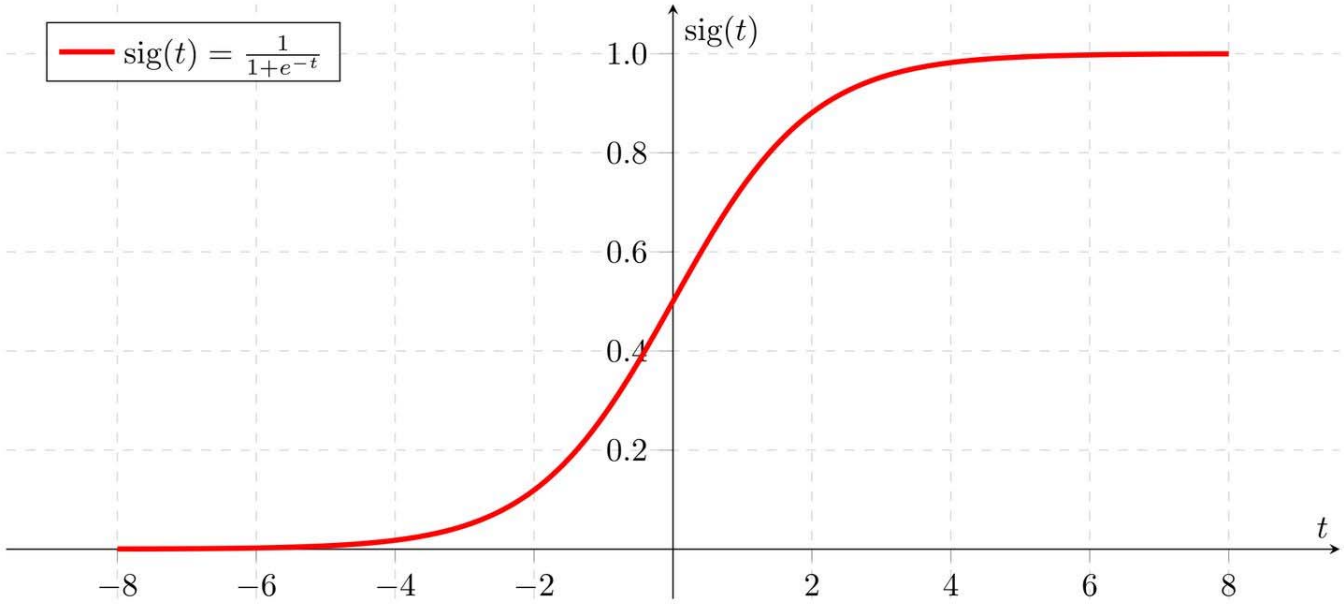
- 神经网络中常用的激活函数

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\sigma'(x) = \sigma(x)\big(1-\sigma(x)\big)$$

$$[log\sigma(x)]' = 1-\sigma(x)$$

$$[\log(1-\sigma(x))]' = \sigma(x)$$



逻辑回归

- 二分类问题
  - 例如：垃圾邮件识别，信用卡欺诈行为识别。
  - Hypothesis函数：

$$h_\theta(\boldsymbol{x}) = \delta(\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n)$$

$$h_\theta(\boldsymbol{x}) = \delta(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

  - 取阈值为0.5，二分类的判别公式为：

$$y(\boldsymbol{x}) = \begin{cases} 1, & \text{如果} h_\theta(\boldsymbol{x}) \geq 0.5 \\ 0, & \text{如果} h_\theta(\boldsymbol{x}) < 0.5 \end{cases}$$



逻辑回归

- 输出激活函数是 sigmoid 函数，值域为 0 ~ 1。模型输出值：标签为1的概率。
- 模型训练参数 $\boldsymbol{\theta}$，使得似然函数最大：

$$p(y(\boldsymbol{x_1}) = y_1 \mid \boldsymbol{\theta}, \boldsymbol{x_1}) * p(y(\boldsymbol{x_2}) = y_2 \mid \boldsymbol{\theta}, \boldsymbol{x_2}) * \cdots * p(y(\boldsymbol{x_m}) = y_m \mid \boldsymbol{\theta}, \boldsymbol{x_m})$$

取 log 为：$\sum_{k=1}^{m} \log(p(y(\boldsymbol{x_k}) = y_k \mid \boldsymbol{\theta}, \boldsymbol{x_k}))$

$$p(y(\boldsymbol{x_k}) = y_k \mid \boldsymbol{\theta}, \boldsymbol{x_k}) = \begin{cases} h_\theta(\boldsymbol{x_k}), & \text{如果} y_k = 1 \\ 1 - h_\theta(\boldsymbol{x_k}), & \text{如果} y_k = 0 \end{cases}$$

$$p(y(\boldsymbol{x_k}) = y_k \mid \boldsymbol{\theta}, \boldsymbol{x_k}) = y_k * h_\theta(\boldsymbol{x_k}) + (1 - y_k)(1 - h_\theta(\boldsymbol{x_k}))$$

整体损失函数：$\boldsymbol{J(\theta)} = -\frac{1}{m}\sum_{k=1}^{m}(y_k * h_\theta(\boldsymbol{x_k}) + (1 - y_k)(1 - h_\theta(\boldsymbol{x_k})))$

Bayes公式

- $P(A)$：事件A 发生的概率，$P(B)$：事件B 发生的概率。
- $P(A \mid B)$：事件B 发生的条件下，事件A 发生的概率。
- $P(A, B)$：事件A 和事件B 同时发生的概率。

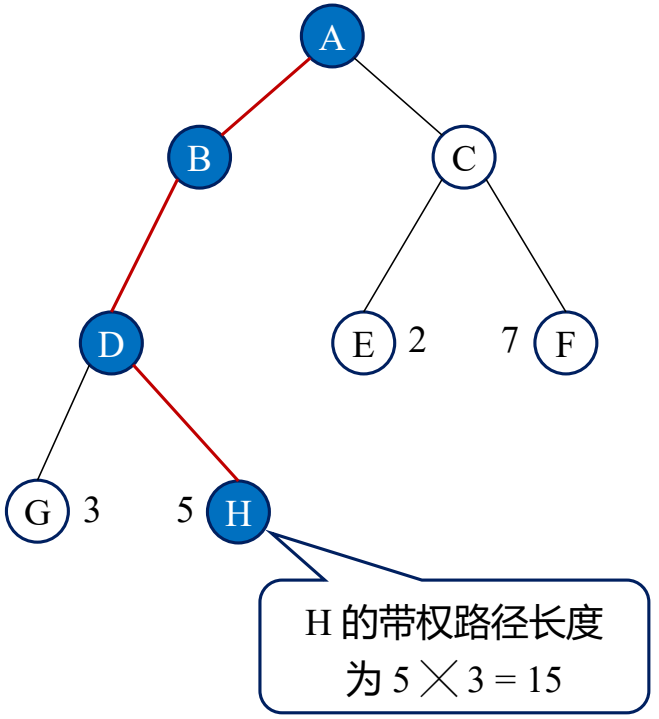
$$P(A \mid B) = \frac{P(A,B)}{P(B)}, \quad P(B \mid A) = \frac{P(A,B)}{P(A)}$$

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

Huffman编码

- 数据结构：树
  - 路径和路径长度
  - 节点的权和带权路径长度
  - 树的带权路径长度

    所有叶子节点的带权路径之和。
- Huffman树：最优二叉树。给定 n 个权值作为叶子节点，带权路径长度最小的树。



Huffman编码

- Huffman树的构造

  给定 n 个权值$\{w_1, w_2, \ldots, w_n\}$ 作为二叉树的叶子节点

  - 将$\{w_1, w_2, \ldots, w_n\}$ 看成有 n 棵树的森林（每棵树只有一个节点）。
  - 选出根节点权值最小的树合并，新树的根节点权值为两棵树权值之和。
  - 从森林里删除两棵树，将新树加入森林。
  - 重复（2）（3）步直到森林只剩下一棵树，该数即为哈夫曼树。

Huffman编码

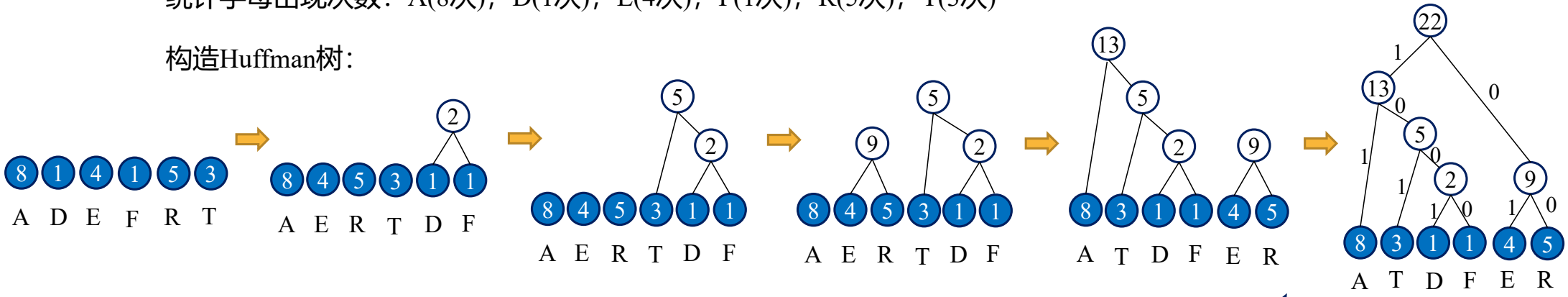
北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

Huffman编码的构造-示例

传送报文“AFTER DATA EAR ARE ART AREA”

统计字母出现次数：A(8次)，D(1次)，E(4次)，F(1次)，R(5次)，T(3次)

构造Huffman树：



编码：A（11），D（1001），E（01），F（1000），R（00），T（101）

此种编码方式
传送报文，是否优于
（3位数字）等长编码？

背景知识

统计语言模型

- 统计语言模型：计算语句出现概率的**概率模型**。

假设语句 $W = w_{1\sim T} \coloneqq (w_1, w_2, \dots, w_T)$，组成该语句的词语的联合概率为：

$$p(W) = p(w_1, w_2, \dots, w_T) = p(w_1) * p(w_2 \mid w_1) * p(w_3 \mid w_{1\sim 2}) * \cdots * p(w_T \mid w_{1\sim(T-1)})$$

条件概率：$p(w_1)$，$p(w_2 \mid w_1)$，$p(w_3 \mid w_{1\sim 2})$，…，$p(w_T \mid w_{1\sim(T-1)})$ 是模型的参数。

当语料库足够大时：$p(w_k \mid w_{1\sim(k-1)}) = \frac{p(w_1,\dots,w_k)}{p(w_1,\dots,w_{k-1})} \approx \frac{count(w_1,\dots,w_k)}{count(w_1,\dots,w_{k-1})}$

Markov假设，一个词的出现只跟前面*n-1* 个词有关：

$$p(w_k \mid w_{1\sim(k-1)}) \approx \frac{p(w_{k-n+1}, \dots, w_k)}{p(w_{k-n+1}, \dots, w_{k-1})} \approx \frac{count(w_{k-n+1}, \dots, w_k)}{count(w_{k-n+1}, \dots, w_{k-1})}$$

n-gram
语言模型

统计语言模型

- n-gram语言模型的参数数量。

n	模型参数数量
1 (unigram)	2×10^5
2 (bigram)	4×10^{10}
3 (trigram)	8×10^{15}
4 (4-gram)	16×10^{20}

词典大小

N = 200000

统计语言模型

- 模型的目标函数：

$$\prod_{w \in C} p(w|Context(w))$$

— C 表示语料库

— Context(w) 表示 w 的上下文

— 对于 n-gram 模型，Context(w) = $w_{i-n+1 \sim i-1}$

统计语言模型

- 采用做大似然概率的目标函数：

$$L = \sum_{w \in C} \log(p(w|Context(w)))$$

$p(w|Context(w))$是关于 w 和 Context(w) 的函数，即：

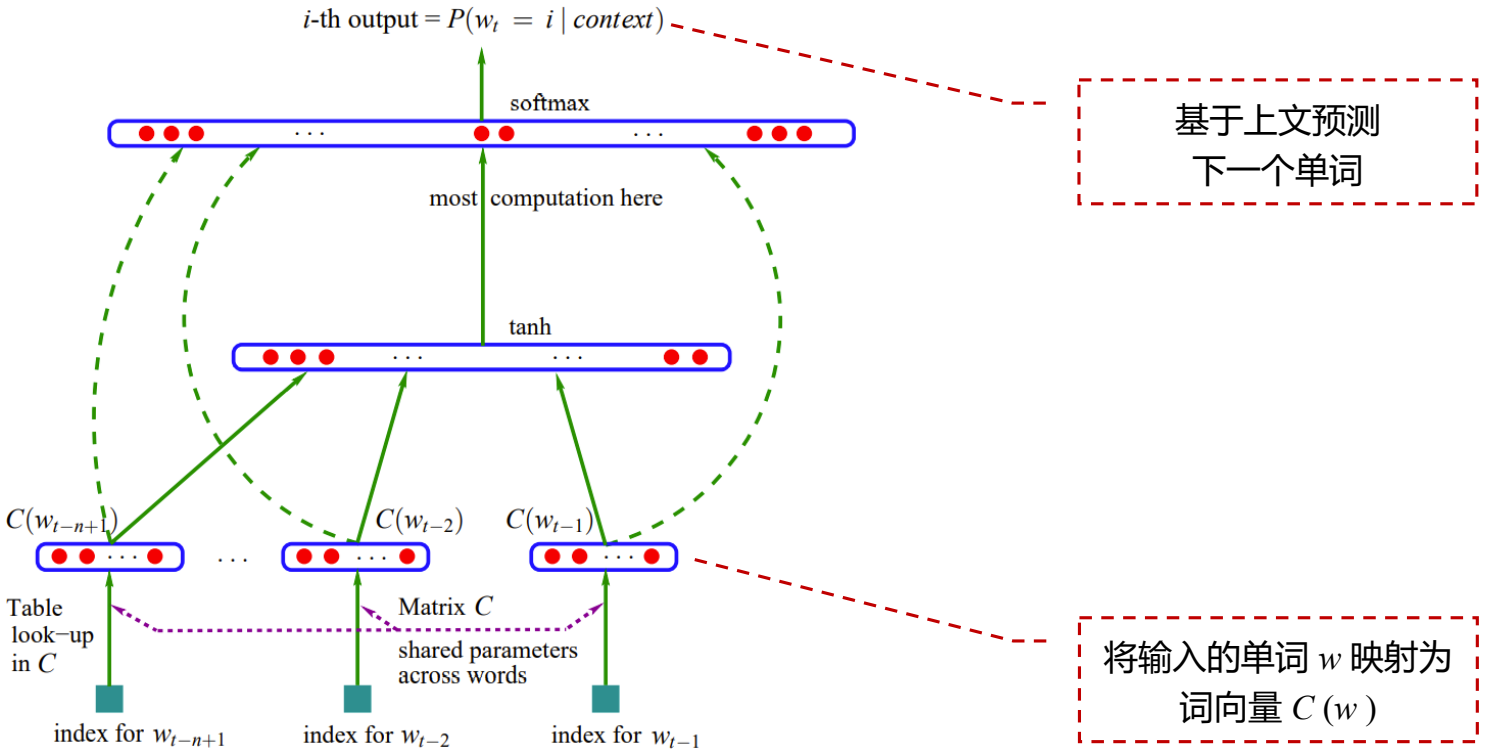
$$p(w|Context(w)) = F(w, Context(w), \theta)$$

θ 为待定参数集。

难点：
1. 函数F的构造
2. 最优参数的求解

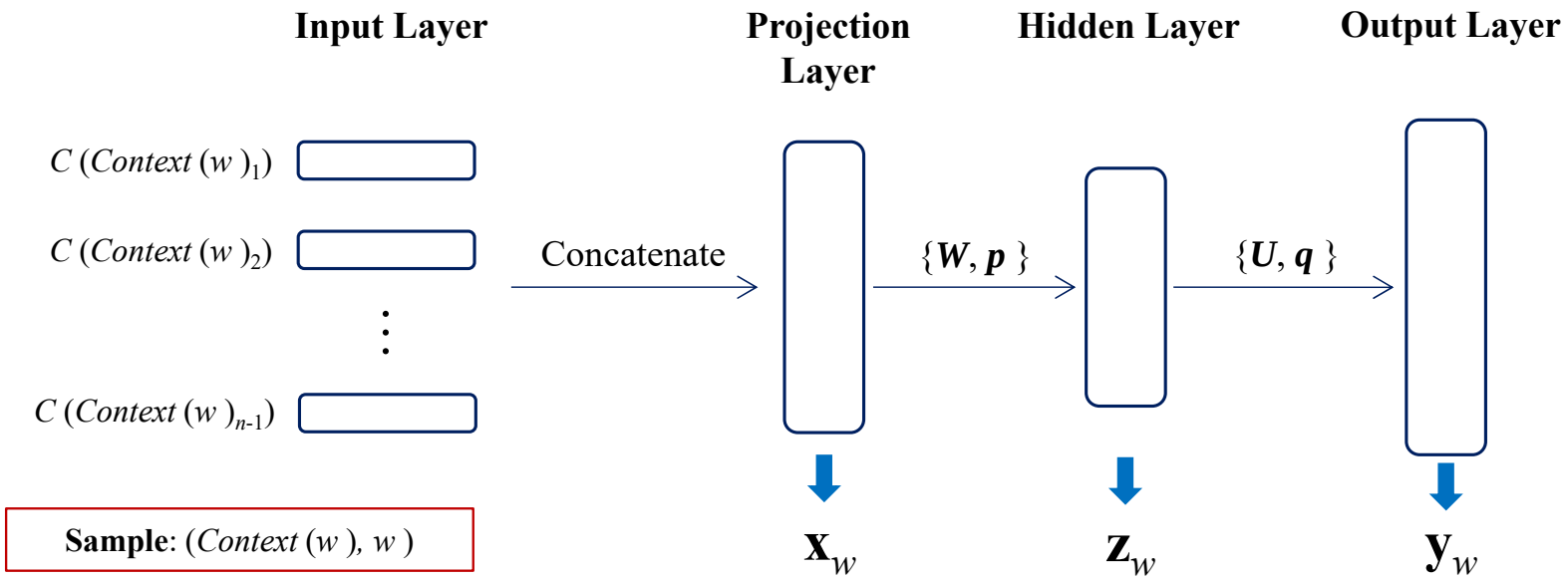
神经概率语言模型

- Bengio et. al: “A Neural Probabilistic Language Model”



神经概率语言模型

- Bengio et. al: “A Neural Probabilistic Language Model”



$$\mathbf{z}_w = \tanh(\boldsymbol{W} \cdot \mathbf{x}_w + \boldsymbol{p})$$

$$\mathbf{y}_w = \boldsymbol{U} \cdot \mathbf{z}_w + \boldsymbol{q}$$

神经概率语言模型

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

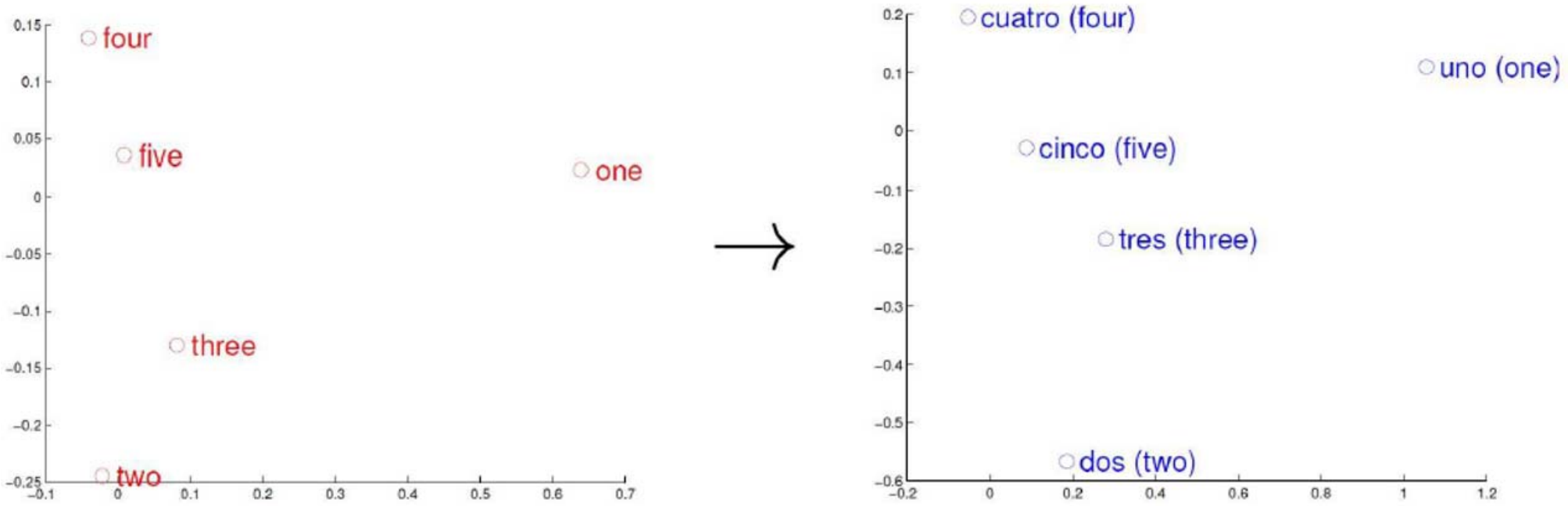
- Bengio et. al: “A Neural Probabilistic Language Model”
  - 输出 $\mathbf{y}_w = (y_{w,1}, y_{w,2}, \ldots, y_{w,N})^{\mathrm{T}}$

    $$p(w|Context(w)) = \frac{e^{y_{w,i_w}}}{\sum_{i=1}^{N} e^{y_{w,i}}}$$

    $F(w, \mathrm{Context}(w), \theta)$
  - 待确定的参数 θ 包括：
    - 词向量： $\mathbf{v}(\mathrm{w}) \in \mathbb{R}^m$, $w \in \mathcal{D}$
    - 神经网络参数：$\boldsymbol{W} \in \mathbb{R}^{n_h \times (n-1)m}$, $\boldsymbol{p} \in \mathbb{R}^{n_h}$, $\boldsymbol{U} \in \mathbb{R}^{N \times n_h}$, $\boldsymbol{q} \in \mathbb{R}^N$
      - n-1 : *context* 的词语数量，通常不超过5
      - m : 词向量的维数，$10^1 \sim 10^2$
      - n_h : 由用户指定，通常为10^2量级
      - $N = |D|$，即语料的词汇集规模，$10^4 \sim 10^5$

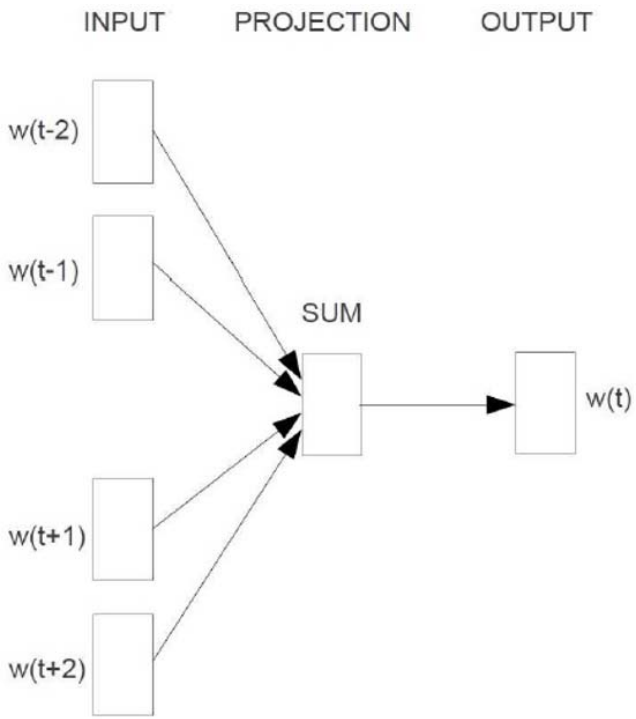
神经概率语言模型

- 词向量的直观理解

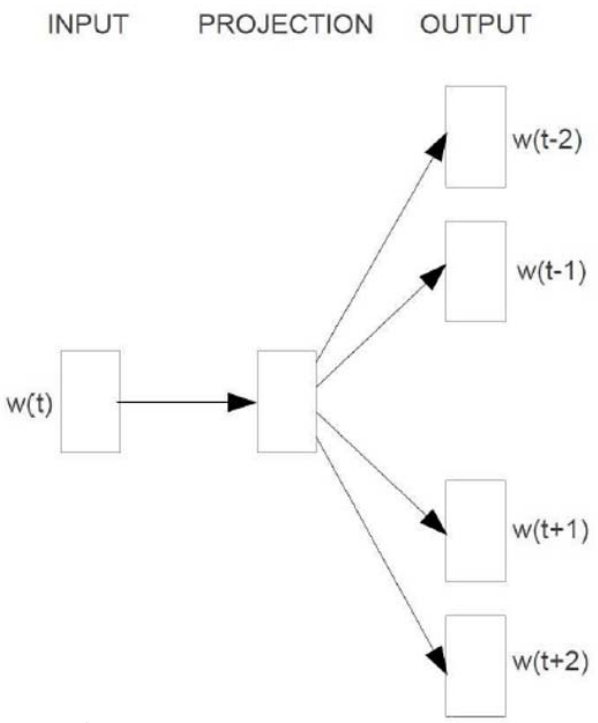


英文和西班牙文的词向量对比（PCA降维形成二维数组）

Word2vec模型

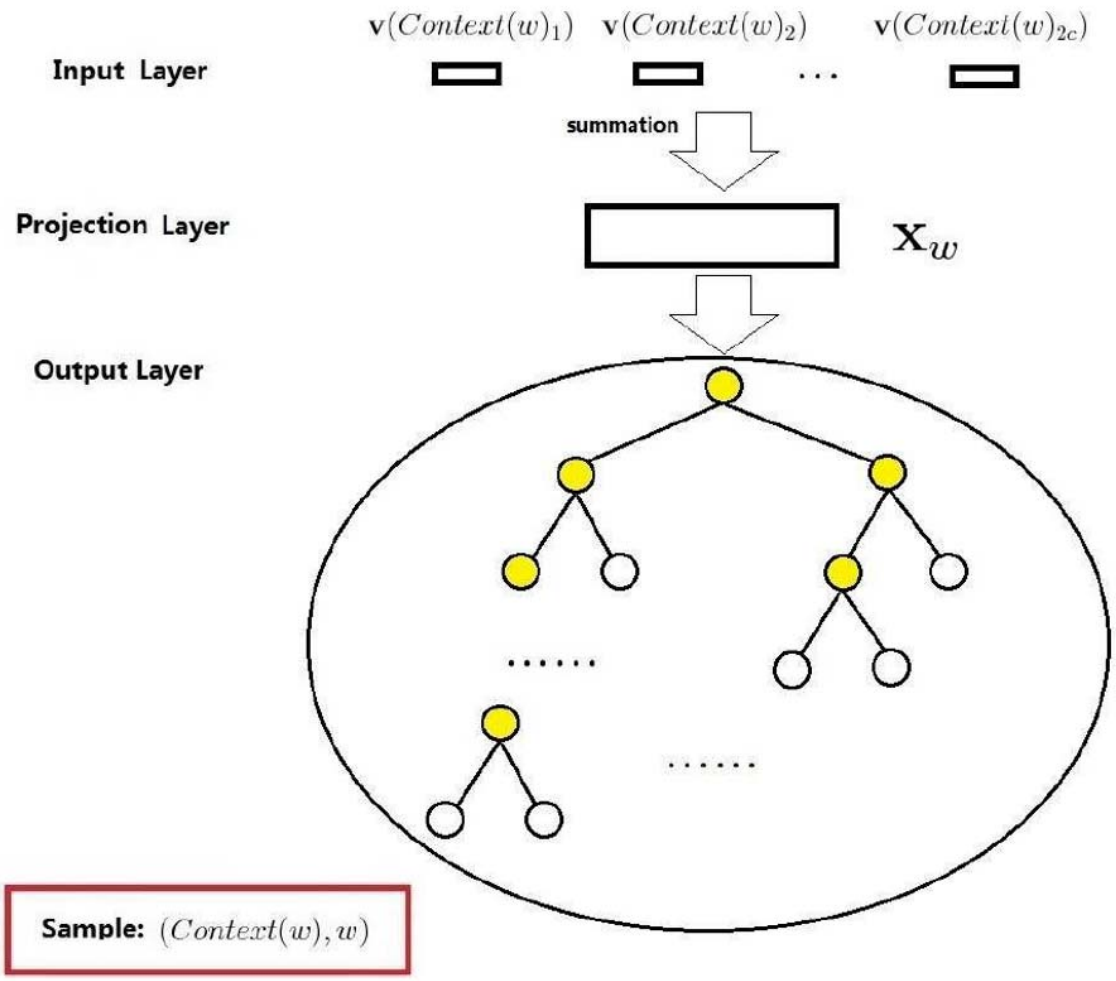


CBOW模型



Skip-gram模型

基于 Hierarchical Softmax 的 CBOW 模型



$2c$ 个词的词向量（维度为 m ）

$$\boldsymbol{x}_w = \sum_{i=1}^{2c} \boldsymbol{v}(Context(w)_i) \in \mathbb{R}^m$$

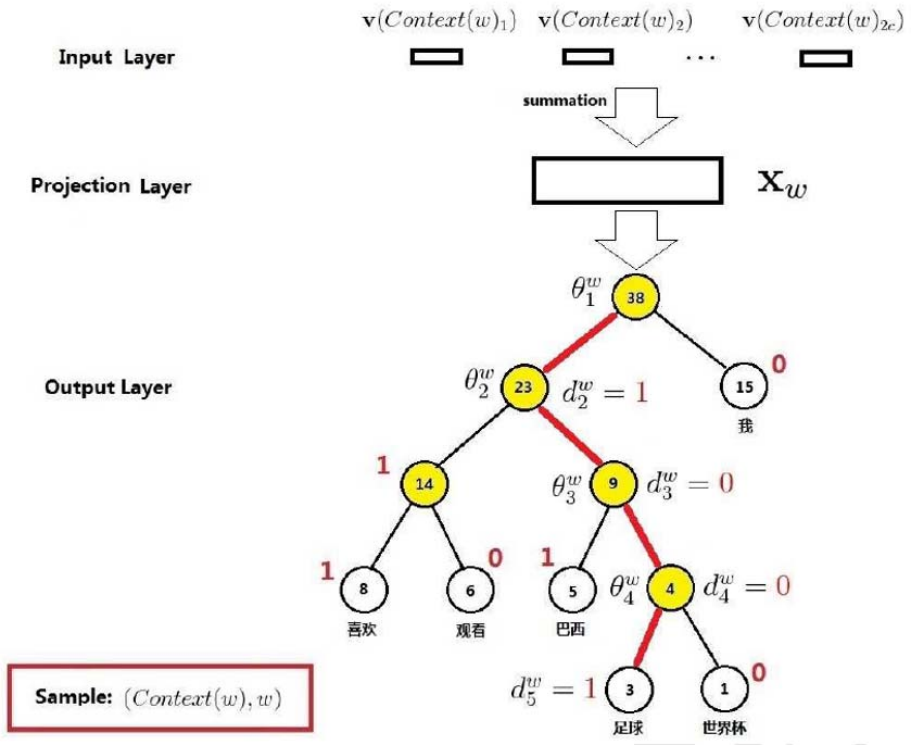
Huffman树：以语料中出现过的词语作为叶子节点，以各词在语料中出现的次数作为权值。

基于 Hierarchical Softmax 的 CBOW 模型

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

Huffman树：

1. p^w：从根节点出发到达词语 w 对应的叶子节点的路径。
2. l^w：路径 p^w 包含的节点的个数。
3. $p_1^w, p_2^w, \ldots, p_{l^w}^w$：路径 p^w 中的 l^w 个节点，其中p_1^w表示根节点，$p_{l^w}^w$表示词 w 对应的节点。
4. $d_2^w, d_3^w, \ldots, d_{l^w}^w \in \{0,1\}$：词 w 的Huffman编码，它由 l^w-1 位编码构成，d_j^w表示路径 p^w 中第j 个节点对应的编码（根节点不对应编码）。
5. $\theta_1^w, \theta_2^w, \ldots, \theta_{l^w-1}^w \in \mathbb{R}^m$：路径 p^w 中非叶子节点对应的向量，θ_j^w 表示路径 p^w 中第 j 个非叶子节点对应的向量。



基于 Hierarchical Softmax 的 CBOW 模型

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

Huffman树：

二分类标签：$Label(p_i^w) = 1 - d_i^w, i = 2,3,\dots,l^w.$

正类（label=1）的概率：$\sigma(\boldsymbol{x}_w^T\theta) = \frac{1}{1+e^{-x_w^T\theta}}$

以词语“足球”为例，每次分类结果的概率：

1. 第1次：$p(d_2^w|x_w,\theta_1^w) = 1 - \sigma(x_w^T\theta_1^w)$；
2. 第2次：$p(d_3^w|x_w,\theta_2^w) = \sigma(x_w^T\theta_2^w)$；
3. 第3次：$p(d_4^w|x_w,\theta_3^w) = \sigma(x_w^T\theta_3^w)$；
4. 第4次：$p(d_5^w|x_w,\theta_4^w) = 1 - \sigma(x_w^T\theta_4^w)$；

$$p(\text{足球}|Context(\text{足球})) = \prod_{j=2}^{5} p(d_j^w|x_w,\theta_{j=1}^w)$$

一般地，

$$p(w|Context(w)) = \prod_{j=2}^{l^w} p(d_j^w|x_w,\theta_{j-1}^w)$$

其中，

$$p(d_j^w|x_w,\theta_{j-1}^w) = \begin{cases} \sigma(x_w^T\theta_{j-1}^w), & d_j^w = 0 \\ 1 - \sigma(x_w^T\theta_{j-1}^w), & d_j^w = 1 \end{cases}$$

即，

$$p(d_j^w|x_w,\theta_{j-1}^w) = [\sigma(x_w^T\theta_{j-1}^w)]^{1-d_j^w} \cdot [1 - \sigma(x_w^T\theta_{j-1}^w)]^{d_j^w}$$

$$\sum_{w\in D} p(w|Context(w)) = 1\,?$$

基于 Hierarchical Softmax 的 CBOW 模型

$$L = \sum_{w \in C} log \prod_{j=2}^{l^w} \{[\sigma(x_w^T \theta_{j-1}^w)]^{1-d_j^w} \cdot [1 - \sigma(x_w^T \theta_{j-1}^w)]^{d_j^w}\}$$

$$= \sum_{w \in C} \sum_{j=2}^{l^w} \{(1 - d_j^w) \cdot log[\sigma(x_w^T \theta_{j-1}^w)] + d_j^w \cdot \log[1 - \sigma(x_w^T \theta_{j-1}^w)]\}$$

设，$L(w,j) = (1 - d_j^w) \cdot log[\sigma(x_w^T \theta_{j-1}^w)] + d_j^w \cdot \log[1 - \sigma(x_w^T \theta_{j-1}^w)]$

求$L(w,j)$关于θ_{j-1}^w的梯度：

$$\frac{\partial L(w,j)}{\partial \theta_{j-1}^w} = \frac{\partial}{\partial \theta_{j-1}^w} \{(1 - d_j^w) \cdot log[\sigma(x_w^T \theta_{j-1}^w)] + d_j^w \cdot \log[1 - \sigma(x_w^T \theta_{j-1}^w)]\}$$

$$= (1 - d_j^w)[1 - \sigma(x_w^T \theta_{j-1}^w)] x_w - d_j^w \sigma(x_w^T \theta_{j-1}^w) x_w$$

$$= \{(1 - d_j^w)[1 - \sigma(x_w^T \theta_{j-1}^w)] - d_j^w \sigma(x_w^T \theta_{j-1}^w)\} x_w = [1 - d_j^w - \sigma(x_w^T \theta_{j-1}^w)] x_w$$

θ_{j-1}^w的更新：

$$\theta_{j-1}^w := \theta_{j-1}^w + \eta[1 - d_j^w - \sigma(x_w^T \theta_{j-1}^w)] x_w$$

基于 Hierarchical Softmax 的 CBOW 模型

同理，求$L(w,j)$关于 x_w 的梯度：

$$\frac{\partial L(w,j)}{\partial x_w} = [1 - d_j^w - \sigma(x_w^T \theta_{j-1}^w)]\theta_{j-1}^w$$

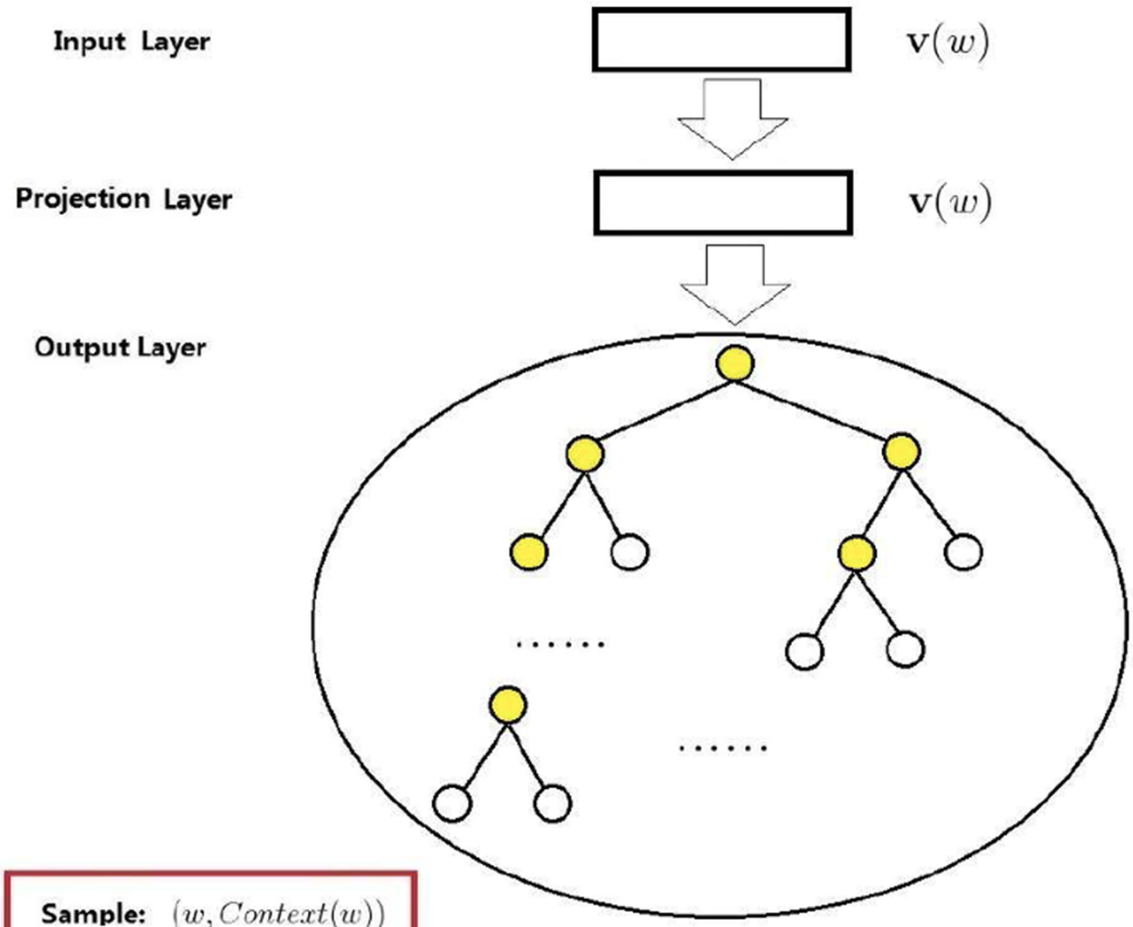
词向量的更新：

$$\mathrm{v}(\tilde{w}) := \mathrm{v}(\tilde{w}) + \eta \sum_{j=2}^{l^w} \frac{\partial L(w,j)}{\partial x_w}, \tilde{w} \in Context(w)$$

采用梯度上升法更新参数的伪代码：

1. $\mathbf{e} = 0$.
2. $\mathbf{x}_w = \sum_{u \in Context(w)} \mathbf{v}(u)$.
3. **FOR** $j = 2 : l^w$ **DO**
   {
   3.1 $q = \sigma(\mathbf{x}_w^\top \theta_{j-1}^w)$
   3.2 $g = \eta(1 - d_j^w - q)$
   3.3 $\mathbf{e} := \mathbf{e} + g\theta_{j-1}^w$
   3.4 $\theta_{j-1}^w := \theta_{j-1}^w + g\mathbf{x}_w$
   }
4. **FOR** $u \in Context(w)$ **DO**
   {
   $\mathbf{v}(u) := \mathbf{v}(u) + \mathbf{e}$
   }

基于 Hierarchical Softmax 的 Skip-gram 模型



词 w 的词向量 $\mathrm{v}(w) \in \mathbb{R}^m$

Huffman树：以语料中出现过的词语作为叶子节点，以各词在语料中出现的次数作为权值。

基于 Hierarchical Softmax 的 Skip-gram 模型

目标函数：$p(Context(w)|w) = \prod_{u \in Context(w)} p(u|w)$

其中，$p(u|w) = \prod_{j=2}^{l^u} p(d_j^u | v(w), \theta_{j-1}^u)$

$$p\left(d_j^u \middle| v(w), \theta_{j-1}^u\right) = [\sigma(v(w)^T \theta_{j-1}^u)]^{1-d_j^u} \cdot [1 - \sigma(v(w)^T \theta_{j-1}^u)]^{d_j^u}$$

$$L = \sum_{w \in C} log \prod_{u \in Context(w)} \prod_{j=2}^{l^u} \{[\sigma(v(w)^T \theta_{j-1}^u)]^{1-d_j^u} \cdot [1 - \sigma(v(w)^T \theta_{j-1}^u)]^{d_j^u}\}$$

$$= \sum_{w \in C} \sum_{u \in Context(w)} \sum_{j=2}^{l^u} \{\left(1 - d_j^u\right) \cdot log\left[\sigma\left(v(w)^T \theta_{j-1}^u\right)\right] + d_j^u \cdot log[1 - \sigma(v(w)^T \theta_{j-1}^u)]\}$$

令，$L(w, u, j) = \left(1 - d_j^u\right) \cdot log\left[\sigma\left(v(w)^T \theta_{j-1}^u\right)\right] + d_j^u \cdot log[1 - \sigma(v(w)^T \theta_{j-1}^u)]$

基于 Hierarchical Softmax 的 Skip-gram 模型

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

$L(w,u,j)$ 关于 θ_{j-1}^u 的梯度：

$$\frac{\partial L(w,u,j)}{\partial \theta_{j-1}^u} = \frac{\partial}{\partial \theta_{j-1}^u}\{(1-d_j^u)\cdot log[\sigma(v(w)^T\theta_{j-1}^u)] + d_j^u \cdot log[1-\sigma(v(w)^T\theta_{j-1}^u)]\}$$

$$= (1-d_j^u)[1-\sigma(v(w)^T\theta_{j-1}^u)]\mathrm{v}(w) - d_j^u\sigma(v(w)^T\theta_{j-1}^u)v(w)$$

$$= \{(1-d_j^u)[1-\sigma(v(w)^T\theta_{j-1}^u)] - d_j^u\sigma(v(w)^T\theta_{j-1}^u)\}\, v(w)$$

$$= [1-d_j^u-\sigma(v(w)^T\theta_{j-1}^u)]\, v(w)$$

θ_{j-1}^u 的更新公式为：$\theta_{j-1}^u := \theta_{j-1}^u + \eta[1-d_j^u-\sigma(v(w)^T\theta_{j-1}^u)]\, v(w)$

同理，$L(w,u,j)$ 关于 $\mathrm{v}(w)$ 的梯度：$\frac{\partial L(w,u,j)}{\partial v(w)} = [1-d_j^u-\sigma(v(w)^T\theta_{j-1}^u)]\theta_{j-1}^u$

$\mathrm{v}(w)$ 的更新公式为：$v(w) := v(w) + \eta\sum_{u\in Context(w)}\sum_{j=2}^{l^u}\frac{\partial L(w,u,j)}{\partial v(w)}$

基于 Hierarchical Softmax 的 Skip-gram 模型

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

采用梯度上升法更新参数的伪代码：

$\mathbf{e} = \mathbf{0}$
FOR $u \in Context(w)$ **DO**
{
  FOR $j = 2 : l^u$ **DO**
  {
    1. $q = \sigma(\mathbf{v}(w)^\top \theta^u_{j-1})$
    2. $g = \eta(1 - d^u_j - q)$
    3. $\mathbf{e} := \mathbf{e} + g\theta^u_{j-1}$
    4. $\theta^u_{j-1} := \theta^u_{j-1} + g\mathbf{v}(w)$
  }
}
$\mathbf{v}(w) := \mathbf{v}(w) + \mathbf{e}$

每处理一个词更新一次参数：

FOR $u \in Context(w)$ **DO**
{
  $\mathbf{e} = \mathbf{0}$
  FOR $j = 2 : l^u$ **DO**
  {
    1. $q = \sigma(\mathbf{v}(w)^\top \theta^u_{j-1})$
    2. $g = \eta(1 - d^u_j - q)$
    3. $\mathbf{e} := \mathbf{e} + g\theta^u_{j-1}$
    4. $\theta^u_{j-1} := \theta^u_{j-1} + g\mathbf{v}(w)$
  }
  $\mathbf{v}(w) := \mathbf{v}(w) + \mathbf{e}$
}

基于 Negative Sampling的CBOW模型

关于 w 的负样本子集 $NEG(w) \neq \emptyset$. 且对 $\forall \tilde{w} \in D$，定义：

$$L^w(\tilde{w}) = \begin{cases} 1, & \tilde{w} = w \\ 0, & \tilde{w} \neq w \end{cases}$$

对于给定的正样本 (Context(w), w)，模型目标最大化：

$$g(w) = \prod_{u \in \{w, NEG(w)\}} p(u|Context(w))$$

其中，$p(u|Context(w)) = \begin{cases} \sigma(x_w^T \theta^u), & L^w(u) = 1 \\ 1 - \sigma(x_w^T \theta^u), & L^w(u) = 0 \end{cases}$

x_w：Context(w)各个词的词向量之和

θ^u：待训练的参数

$$p(u|Context(w)) = [\sigma(x_w^T \theta^u)]^{L^w(u)} \cdot [1 - \sigma(x_w^T \theta^u)]^{1 - L^w(u)}$$

$$g(w) = \sigma(x_w^T \theta^w) \cdot \prod_{u \in NEG(w)} [1 - \sigma(x_w^T \theta^u)]$$

增大正样本 $\sigma(x_w^T \theta^w)$ 的概率，减小负样本 $\sigma(x_w^T \theta^u)$ 的概率

基于 Negative Sampling的CBOW模型

整体优化的目标：$G = \prod_{w \in C} g(w)$

目标函数：$L = \log G = log \prod_{w \in C} g(w) = \sum_{w \in C} \log g(w)$

$$= \sum_{w \in C} \log \prod_{u \in \{w, NEG(w)\}} \{[\sigma(x_w^T \theta^u)]^{L^w(u)} \cdot [1 - \sigma(x_w^T \theta^u)]^{1 - L^w(u)}\}$$

$$= \sum_{w \in C} \sum_{u \in \{w, NEG(w)\}} \{L^w(u) \cdot \log[\sigma(x_w^T \theta^u)] + [1 - L^w(u)] \cdot \log[1 - \sigma(x_w^T \theta^u)]\}$$

令，$L(w, u) = L^w(u) \cdot \log[\sigma(x_w^T \theta^u)] + [1 - L^w(u)] \cdot \log[1 - \sigma(x_w^T \theta^u)]$

基于 Negative Sampling的CBOW模型

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

$\partial L(w,u)$关于的θ^u梯度:

$$\begin{aligned}\frac{\partial \mathcal{L}(w,u)}{\partial \theta^u} &= \frac{\partial}{\partial \theta^u}\left\{L^w(u)\cdot \log\left[\sigma(\mathbf{x}_w^\top \theta^u)\right] + [1-L^w(u)]\cdot \log\left[1-\sigma(\mathbf{x}_w^\top \theta^u)\right]\right\} \\ &= L^w(u)[1-\sigma(\mathbf{x}_w^\top \theta^u)]\mathbf{x}_w - [1-L^w(u)]\sigma(\mathbf{x}_w^\top \theta^u)\mathbf{x}_w \\ &= \left\{L^w(u)[1-\sigma(\mathbf{x}_w^\top \theta^u)] - [1-L^w(u)]\sigma(\mathbf{x}_w^\top \theta^u)\right\}\mathbf{x}_w \\ &= \left[L^w(u)-\sigma(\mathbf{x}_w^\top \theta^u)\right]\mathbf{x}_w.\end{aligned}$$

θ^u的更新公式: $\theta^u := \theta^u + \eta\left[L^w(u)-\sigma(\mathbf{x}_w^\top \theta^u)\right]\mathbf{x}_w.$

$L(w,u)$ 关于 x_w 的梯度: $\dfrac{\partial \mathcal{L}(w,u)}{\partial \mathbf{x}_w} = \left[L^w(u)-\sigma(\mathbf{x}_w^\top \theta^u)\right]\theta^u.$

$v(\tilde{w}), \tilde{w} \in Context(w)$的更新公式: $\mathbf{v}(\tilde{w}) := \mathbf{v}(\tilde{w}) + \eta \sum_{u\in\{w\}\cup NEG(w)} \dfrac{\partial \mathcal{L}(w,u)}{\partial \mathbf{x}_w}, \quad \tilde{w} \in Context(w).$

基于 Negative Sampling的CBOW模型

采用梯度上升法

更新参数的伪代码：

1. $\mathbf{e} = 0$.
2. $\mathbf{x}_w = \sum_{u \in Context(w)} \mathbf{v}(u)$.
3. **FOR** $u = \{w\} \cup NEG(w)$ **DO**
   {
   3.1 $q = \sigma(\mathbf{x}_w^\top \theta^u)$
   3.2 $g = \eta(L^w(u) - q)$
   3.3 $\mathbf{e} := \mathbf{e} + g\theta^u$
   3.4 $\theta^u := \theta^u + g\mathbf{x}_w$
   }
4. **FOR** $u \in Context(w)$ **DO**
   {
   $\mathbf{v}(u) := \mathbf{v}(u) + \mathbf{e}$
   }

基于 Negative Sampling的Skip-gram模型

目标函数：$G=\prod_{w\in C}\prod_{u\in Context(w)}g(u)=\prod_{w\in C}\prod_{u\in Context(w)}\prod_{z\in\{u,NEG(u)\}}p(z|w)$

其中：$p(z|w)=\begin{cases}\sigma(v(w)^T\theta^z), & L^u(z)=1\\ 1-\sigma(v(w)^T\theta^z), & L^u(z)=0\end{cases}$，即：$p(z|w)=[\sigma(v(w)^T\theta^z)]^{L^u(z)}+[1-\sigma(v(w)^T\theta^z)]^{1-L^u(z)}$

取对数：$L=logG=log\prod_{w\in C}\prod_{u\in Context(w)}g(u)=\sum_{w\in C}\sum_{u\in Context(w)}\log g(u)$

$=\sum_{w\in C}\sum_{u\in Context(w)}\log\prod_{z\in\{u,NEG(u)\}}p(z|w)$

$=\sum_{w\in C}\sum_{u\in Context(w)}\sum_{z\in\{u,NEG(u)\}}\log p(z|w)$

$=\sum_{w\in C}\sum_{u\in Context(w)}\sum_{z\in\{u,NEG(u)\}}\log\{[\sigma(v(w)^T\theta^z)]^{L^u(z)}+[1-\sigma(v(w)^T\theta^z)]^{1-L^u(z)}\}$

$=\sum_{w\in C}\sum_{u\in Context(w)}\sum_{z\in\{u,NEG(u)\}}\{L^u(z)\cdot\log[\sigma(v(w)^T\theta^z)]+[1-L^u(z)]\cdot\log[1-\sigma(v(w)^T\theta^z)]\}$

基于 Negative Sampling的Skip-gram模型

对于给定样本(w, Context(w))，目标函数：

$$g(w) = \prod_{\tilde{w} \in Context(w)} \prod_{u \in \{w\} \cup NEG^{\tilde{w}}(w)} p(u|\tilde{w})$$

每个 $\tilde{w}$ 都有相应的负采样集合

其中：

$$p(u|\tilde{w}) = \begin{cases} \sigma(\mathbf{v}(\tilde{w})^\top \theta^u), & L^w(u) = 1; \\ 1 - \sigma(\mathbf{v}(\tilde{w})^\top \theta^u), & L^w(u) = 0, \end{cases}$$

或者：

$$p(u|\tilde{w}) = \left[\sigma(\mathbf{v}(\tilde{w})^\top \theta^u)\right]^{L^w(u)} \cdot \left[1 - \sigma(\mathbf{v}(\tilde{w})^\top \theta^u)\right]^{1-L^w(u)}$$

基于 Negative Sampling的Skip-gram模型

给定语料库 C，目标函数：$G = \prod_{w \in \mathcal{C}} g(w)$

取对数：

$$\begin{aligned}
\mathcal{L} &= \log G = \log \prod_{w \in \mathcal{C}} g(w) = \sum_{w \in \mathcal{C}} \log g(w) \\
&= \sum_{w \in \mathcal{C}} \log \prod_{\tilde{w} \in Context(w)} \prod_{u \in \{w\} \cup NEG^{\tilde{w}}(w)} \left\{ \left[\sigma(\mathbf{v}(\tilde{w})^\top \theta^u)\right]^{L^w(u)} \cdot \left[1 - \sigma(\mathbf{v}(\tilde{w})^\top \theta^u)\right]^{1-L^w(u)} \right\} \\
&= \sum_{w \in \mathcal{C}} \sum_{\tilde{w} \in Context(w)} \sum_{u \in \{w\} \cup NEG^{\tilde{w}}(w)} \left\{ L^w(u) \cdot \log\left[\sigma(\mathbf{v}(\tilde{w})^\top \theta^u)\right] + \left[1 - L^w(u)\right] \cdot \log\left[1 - \sigma(\mathbf{v}(\tilde{w})^\top \theta^u)\right] \right\}.
\end{aligned}$$

令：$\mathcal{L}(w, \tilde{w}, u) = L^w(u) \cdot \log\left[\sigma(\mathbf{v}(\tilde{w})^\top \theta^u)\right] + \left[1 - L^w(u)\right] \cdot \log\left[1 - \sigma(\mathbf{v}(\tilde{w})^\top \theta^u)\right]$

基于 Negative Sampling的Skip-gram模型

$L(w,\tilde{w},u)$关于θ^u的梯度：

$$\begin{aligned}\frac{\partial \mathcal{L}(w,\tilde{w},u)}{\partial \theta^u} &= \frac{\partial}{\partial \theta^u}\left\{L^w(u)\cdot\log\left[\sigma(\mathbf{v}(\tilde{w})^\top\theta^u)\right]+\left[1-L^w(u)\right]\cdot\log\left[1-\sigma(\mathbf{v}(\tilde{w})^\top\theta^u)\right]\right\}\\ &= L^w(u)[1-\sigma(\mathbf{v}(\tilde{w})^\top\theta^u)]\mathbf{v}(\tilde{w})-[1-L^w(u)]\sigma(\mathbf{v}(\tilde{w})^\top\theta^u)\mathbf{v}(\tilde{w})\\ &= \left\{L^w(u)[1-\sigma(\mathbf{v}(\tilde{w})^\top\theta^u)]-[1-L^w(u)]\sigma(\mathbf{v}(\tilde{w})^\top\theta^u)\right\}\mathbf{v}(\tilde{w})\\ &= \left[L^w(u)-\sigma(\mathbf{v}(\tilde{w})^\top\theta^u)\right]\mathbf{v}(\tilde{w}).\end{aligned}$$

θ^u 的更新公式：$\theta^u := \theta^u + \eta\left[L^w(u)-\sigma(\mathbf{v}(\tilde{w})^\top\theta^u)\right]\mathbf{v}(\tilde{w})$

$L(w,\tilde{w},u)$关于$v(\tilde{w})$的梯度：$\dfrac{\partial \mathcal{L}(w,\tilde{w},u)}{\partial \mathbf{v}(\tilde{w})} = \left[L^w(u)-\sigma(\mathbf{v}(\tilde{w})^\top\theta^u)\right]\theta^u$

$v(\tilde{w})$的更新公式：$\mathbf{v}(\tilde{w}) := \mathbf{v}(\tilde{w}) + \eta\sum\limits_{u\in\{w\}\cup NEG^{\tilde{w}}(w)}\dfrac{\partial \mathcal{L}(w,\tilde{w},u)}{\partial \mathbf{v}(\tilde{w})}$

基于 Negative Sampling的Skip-gram模型

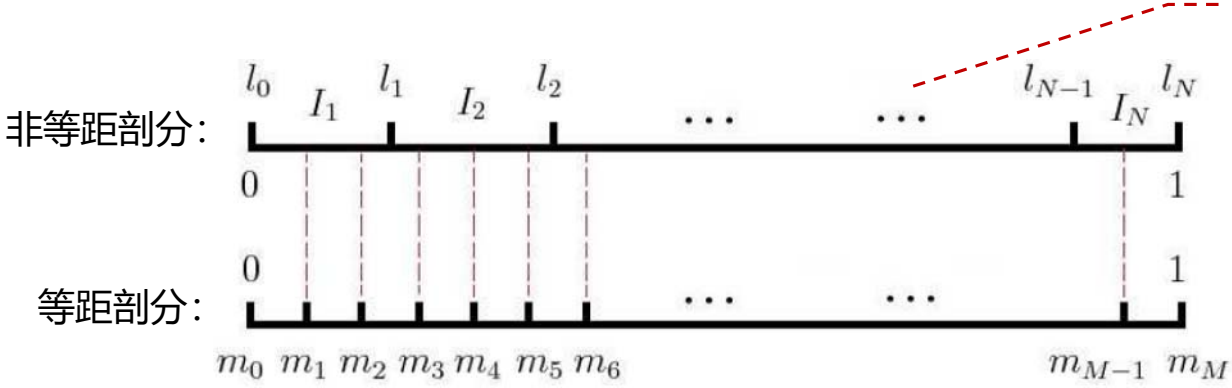
采用随机上升法

更新参数的伪代码:

$$
\begin{aligned}
&\textbf{FOR } \widetilde{w} = Context(w) \textbf{ DO}\\
&\{\\
&\quad \mathbf{e} = \mathbf{0}.\\
&\quad \textbf{FOR } u = \{w\} \cup NEG^{\widetilde{w}}(w) \textbf{ DO}\\
&\quad \{\\
&\qquad q = \sigma(\mathbf{v}(\widetilde{w})^{\top}\theta^u)\\
&\qquad g = \eta(L^w(u) - q)\\
&\qquad \mathbf{e} := \mathbf{e} + g\theta^u\\
&\qquad \theta^u := \theta^u + g\mathbf{v}(\widetilde{w})\\
&\quad \}\\
&\quad \mathbf{v}(\widetilde{w}) := \mathbf{v}(\widetilde{w}) + \mathbf{e}\\
&\}
\end{aligned}
$$

Negative Sampling 负采样算法

给定词语 w，生成 $NEG(w)$.



每个词对应一个线段：

$$I_x = len(w_x) = \frac{count(w_x)}{\sum_{w \in D} count(w)}$$

$M \gg N$

$\{m_j\}_{j=1}^{M-1}$ 与 $\{I_i\}_{i=1}^{N}$ 的映射关系： Table(j) = w_x， j = 1, 2, …, M-1

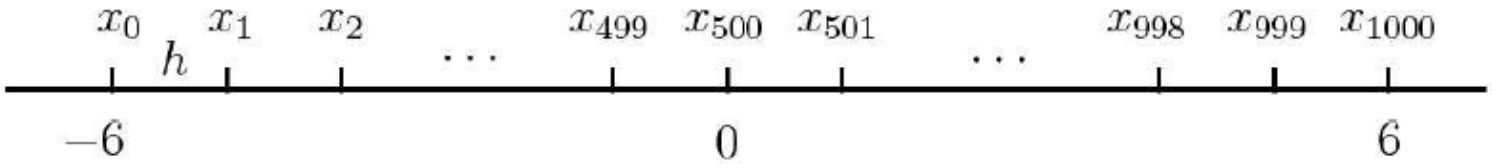
采样过程：每次生成 1~M-1 的一个数字 r，并以 Table(r) 作为一个样本。

word2vec 的实现细节

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

- $\sigma(x)$ 的近似计算

(1) 预先计算 $\sigma(x_i)$ 的值



(2) 查表获取 $\sigma(x)$ 的值

$$\sigma(x) \approx \begin{cases} 0, & x \leq -6 \\ \sigma(x_k), & x \in (-6, 6) \\ 1, & x \geq 6 \end{cases}$$

word2vec 的实现细节

北京雁栖湖应用数学研究院
YANQI LAKE BEIJING INSTITUTE OF MATHEMATICAL SCIENCES AND APPLICATIONS

- 词向量的初始化：

$$\frac{[\text{rand}()/\text{RAND_MAX}] - 0.5}{m}$$

基于上式对向量的每个分量进行初始化。rand()是随机数，m 是词向量的维数。

- 上下文的选取：
  - 设置窗口值 window（默认为5）
  - 生成 [1, window] 区间内的随机数 c
  - 取 w 的前后 c 个词形成 context(w)

问题及讨论

1. 讨论算法时，都是假定语料已经分好词。对于英文语料，分词不是什么问题，但对于中文语料，分词的预处理则是必须的，不同分词器会得到不同的分词结果，那么word2vec所得词向量的质量是如何依赖于中文分词的质量的呢?
2. Hierarchical Softmax 和 Negative Sampling 是两大类不同的方法，但word2vec源码中对这两个分支采用的并不是二择一的方式，用户可以通过参数选取（如取 negative=5，hs=1）同时选中这两支，相当于是一种混合方法，在这种情况下，训练得到的词向量质量如何呢?
3. 投影层采用累积相加的方式，会使得 Context(u)中各词的顺序不再敏感，例如“我爱足球”和“足球爱我”在训练过程中会视为同一样本。因此，如果在投影层换回首尾相接的方式会怎么样呢?
4. word2vec 几个模型中的目标函数中均未考虑正则项，如果加入正则会怎么样呢?
5. 前文介绍的算法都是针对一个静态的语料库来讨论的，如果用户的语料数据是动态阶段性地获取的，譬如，当前只有一批语料，过一段时间后又有另一批语料，…，针对这种情形，如果想进行增量训练，该如何调鳖现有的框架?

问题及讨论

Q&A