Beijing Forest Studio
北京理工大学信息系统及安全对抗实验中心

Deep Learning词向量生成
——CBOW和Skip-gram

王睿怡　　硕士

2017年10月08日

内容提要

- 背景简介
- 基本知识
- 算法原理
- 特点分析
- 应用总结

背景简介

- one-hot Representation

  娃娃：[1 0 0 0 0 0 0 0 0 0 0 0 0 …]

  玩偶：[0 1 0 0 0 0 0 0 0 0 0 0 0 …]

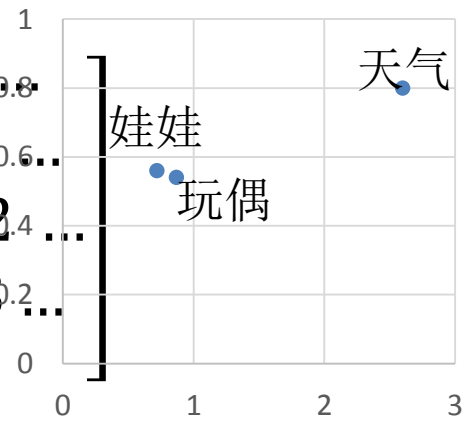
  存在的问题：维数灾难、“词汇鸿沟”现象

- Distributed Representation

  基本思想：通过训练将某种语言中的每一个词映射成一个固定长度的短向量

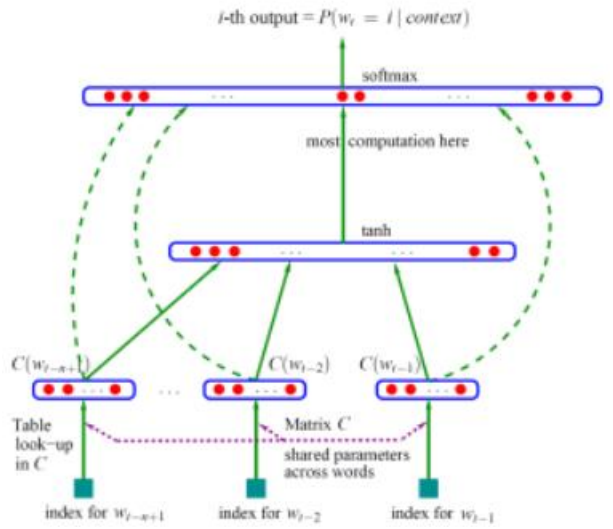
$$\begin{matrix} \text{娃娃} \\ \text{玩偶} \\ \vdots \end{matrix}\begin{bmatrix} 1\,0\,0\,0\,0\,0\ \ldots \\ 0\,1\,0\,0\,0\,0\ \ldots \\ 0\,0\,1\,0\,0\,0\ \ldots \\ 0\,0\,0\,1\,0\,0\ \ldots \\ \vdots \end{bmatrix} \xrightarrow{\text{训练}} \begin{matrix} \text{娃娃} \\ \text{玩偶} \\ \vdots \end{matrix}\begin{bmatrix} 0.81\ 0.54\ 0.23\ -0.24\ 0.32\ \ldots \\ 0.72\ 0.56\ 0.23\ -0.32\ 0.28\ \ldots \\ 0.34\ -0.14\ 0.45\ 0.23\ -0.22\ \ldots \\ -0.33\ 0.55\ 0.23\ -0.35\ 0.43\ \ldots \\ \vdots \end{bmatrix}$$

$V \times V$ $\qquad$ $V \times \mathrm{n}$



研究历史

- 2000，百度IDL的徐伟，神经网络训练语言模型的思想
- 2003，Bengio，神经网络语言模型（NNLM）
- 2008，C&W ， C&W模型
- 2008，M&H ，HLBL模型
- 2014，Mikolov，CBOW模型和Skip-gram模型



基础知识

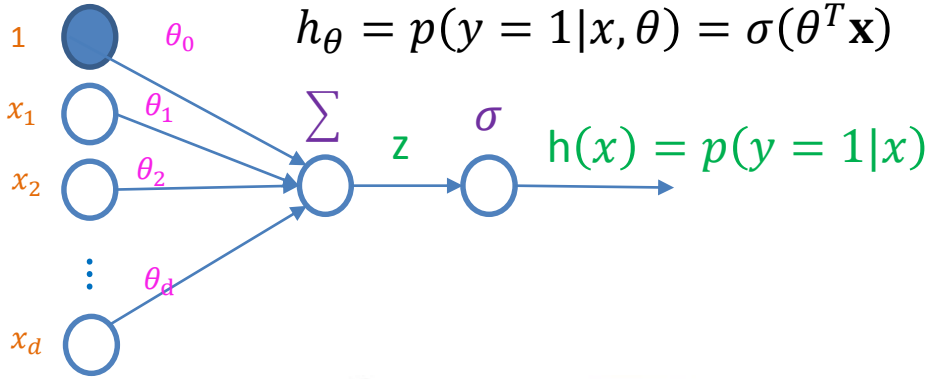
基础知识

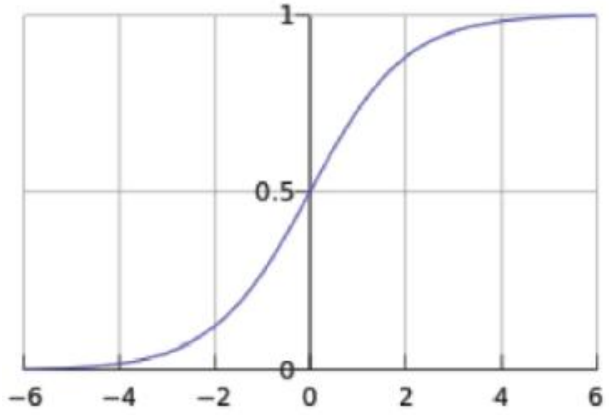
- 交叉熵
  - 度量两个概率分布间的差异性信息
  - 假设有一个样本集中两个概率分布为p、q，其中p为真实分布，q为非真实分布。

  $H(p,q)=\sum_i p(i)\cdot\log(\frac{1}{q(i)})$, 此时将$H(p,q)$称之为交叉熵
  - 交叉熵可在神经网络(机器学习)中作为损失函数
  - 二分类：$H(t,y)=t\cdot log\frac{1}{y}+(1-t)\cdot log\frac{1}{1-y}$　　$t\in\{0,1\}$

基本概念

- logic回归
  - 用于二分类问题





Sigmoid函数

训练样本：$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$

输入特征：$\mathrm{x}^{(i)} \in R^{\mathrm{n}+1}$　类标记：$\mathrm{y}^{(i)} \epsilon \{0,1\}$

假设函数：

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

$$\mathrm{y}(\mathbf{x}) = \begin{cases} 1, & h_\theta(\mathbf{x}) \geq 0.5; \\ 0, & h_\theta(\mathbf{x}) < 0.5. \end{cases}$$

损失函数：

$$J_\theta(x) = -\frac{1}{m}\sum_{i=0}^{m}\left(y^{(i)} log h_\theta\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log(1 - h_\theta(x^{(i)}))\right)$$

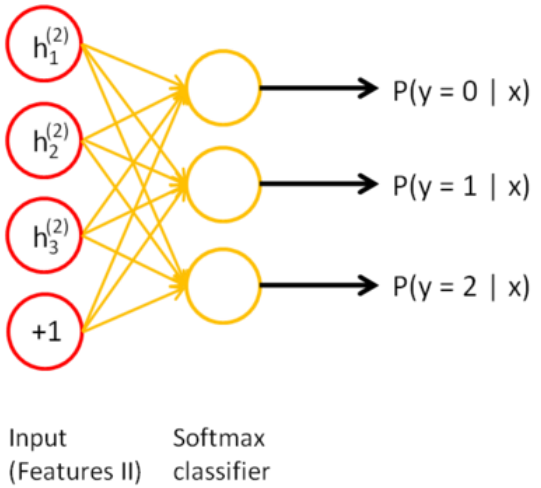
基本知识

- Softmax回归
  - 解决多分类问题
    - 训练集： $\{(x^{(1)}, y^{(1)}), \ldots, (\mathrm{x}^{(m)}, y^{(m)})\}$
    - 类标记： $\mathrm{y}^{(i)} \epsilon \{1,2, \ldots, k\}$
    - 假设函数：

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1|x^{(i)};\theta) \\ p(y^{(i)}=2|x^{(i)};\theta) \\ \vdots \\ p(y^{(i)}=k|x^{(i)};\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

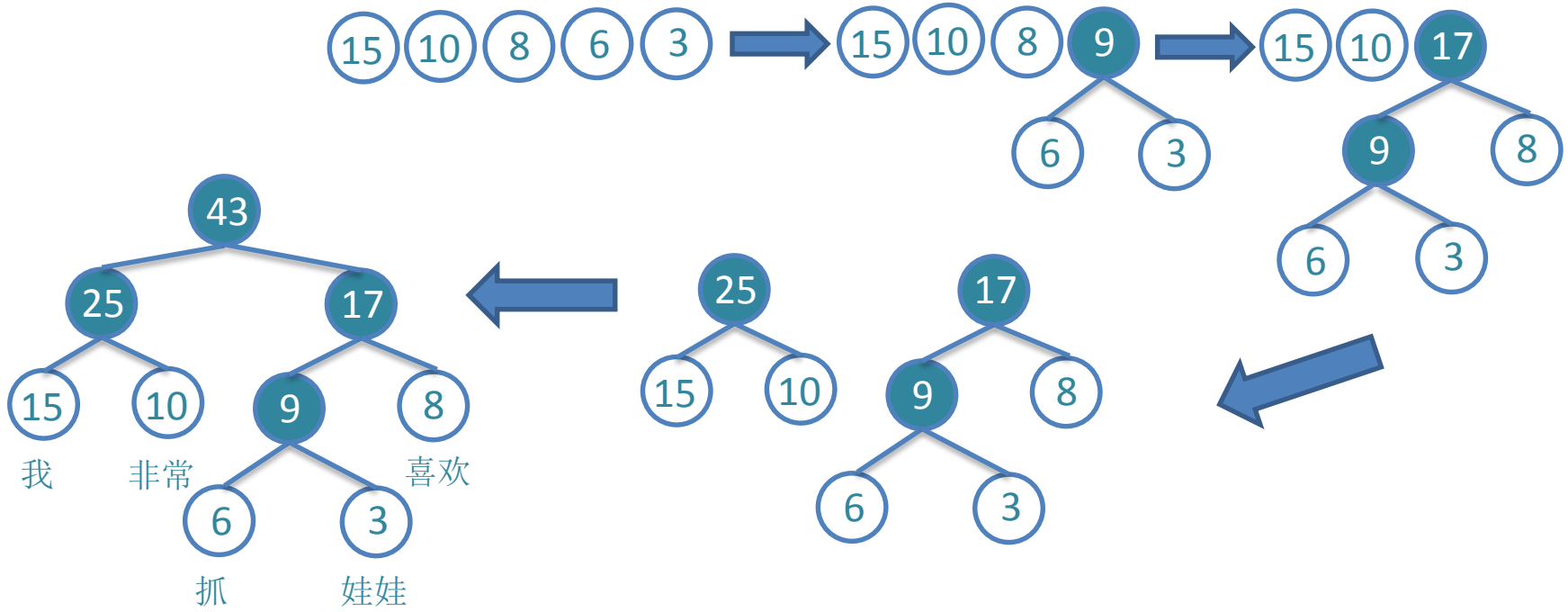
    - 代价函数：

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)}=j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right]$$



基本概念

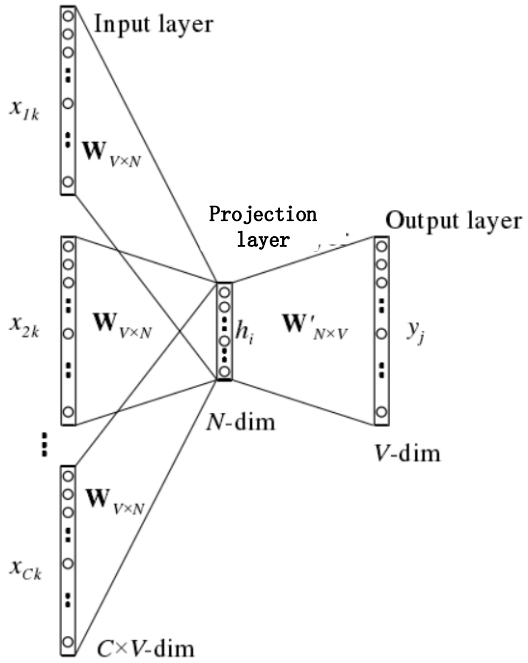
- Huffman树的构造
  - 实例：
    - 在一个文本中，“我”、“非常”、“喜欢”、“抓”、“娃娃”这五个词出现的次数分别是15，10，8，6，3
    - 以这5个词为叶子结点，以相应词频当权值，构造一颗Huffman树



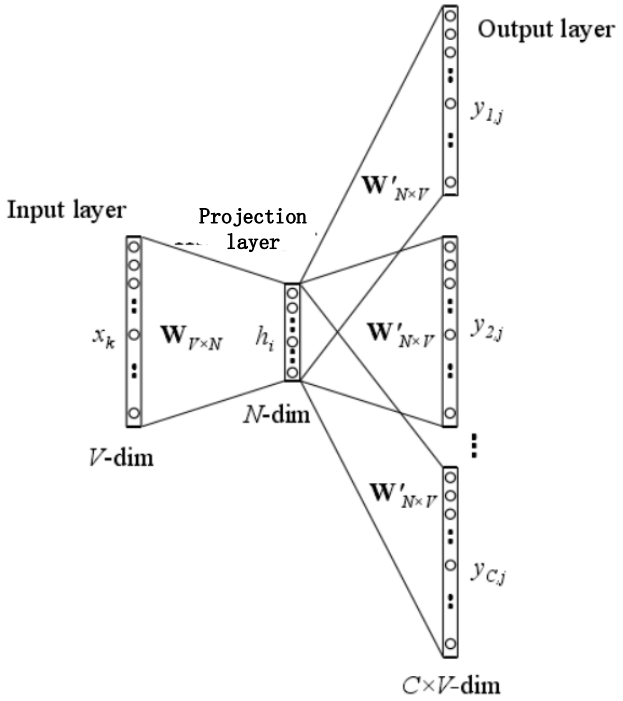
算法原理

算法原理

- Word2vec
  - CBOW模型：用窗口中的词来预测当前词
  - Skip-gram模型：用当前词来预测窗口中的其他词



CBOW模型

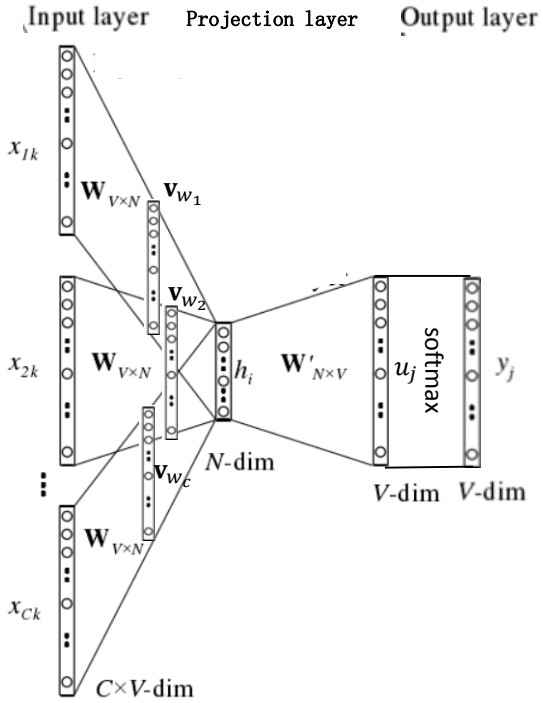


Skip-gram模型

算法原理

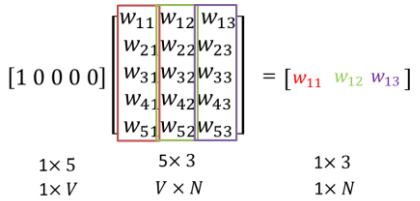
- CBOW模型(Continuous Bag-of- Word Model)
  - 模型的基本思想：用窗口中的词的向量求平均之后来预测中心词。
  - 模型的优化目标：希望预测的概率 y 和真实的中心词one-hot向量 t 一致
  - 任务最终目的：将优化的参数作为词向量的输出结果

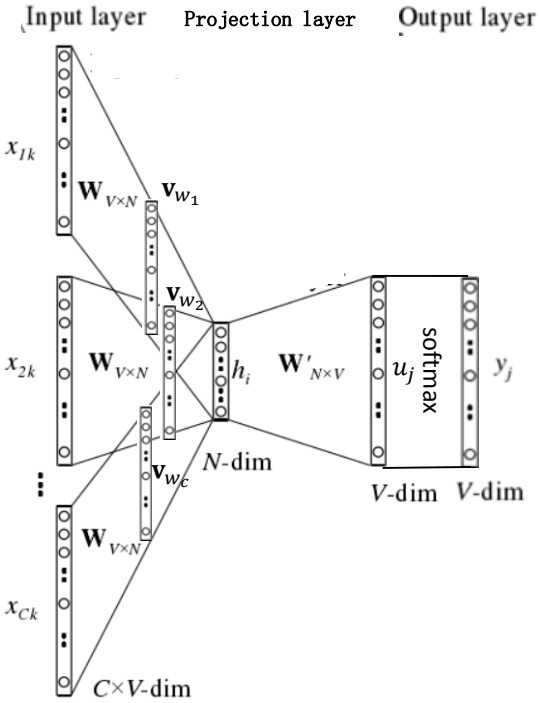
我 非常 喜欢 抓 娃娃



算法原理

- CBOW模型(Continuous Bag-of- Word Model)
  - 利用上下文预测中心词的步骤如下：
    - 大小为C的上下文用one-hot向量表示$(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_C)$
    - $\mathbf{v}_{w_C} = \mathbf{W} \cdot \mathbf{x}_C$
    - $\mathbf{h} = \frac{1}{C}\mathbf{w} \cdot (\mathbf{x}_1 + \mathbf{x}_2 + \cdots \mathbf{x}_C) = \frac{1}{C} \cdot (\mathbf{v}_{w_1} + \mathbf{v}_{w_2} + \cdots \mathbf{v}_{w_C})$
    - $\mathbf{u} = \mathbf{W}' \cdot \mathbf{h}$
    - $\mathbf{y} = softmax(\mathbf{u})$
    - $\mathrm{H}(\mathbf{y}, \mathbf{t}) = -\sum_{j=1}^{|V|} t_j log(y_j) = -t_{j^*} log(y_{j^*}) = -log(y_{j^*})$





- $\min E = -log \mathrm{p}\{w_O | \mathrm{w}_{I,1}, w_{I,2}, \cdots, w_{I,C}\} = -log(y_{j^*}) = -log \frac{exp(u_{j^*})}{\sum_{j'=1}^{V} exp(u_{j'})}$
- $E = -u_{j^*} + log \sum_{j'=1}^{V} exp(u_{j'}) = -\mathbf{v}'^T_{w_O} \cdot \mathrm{h} + log \sum_{j'=1}^{V} exp\left(\mathbf{v}'^T_{w_j} \cdot \mathbf{h}\right)$

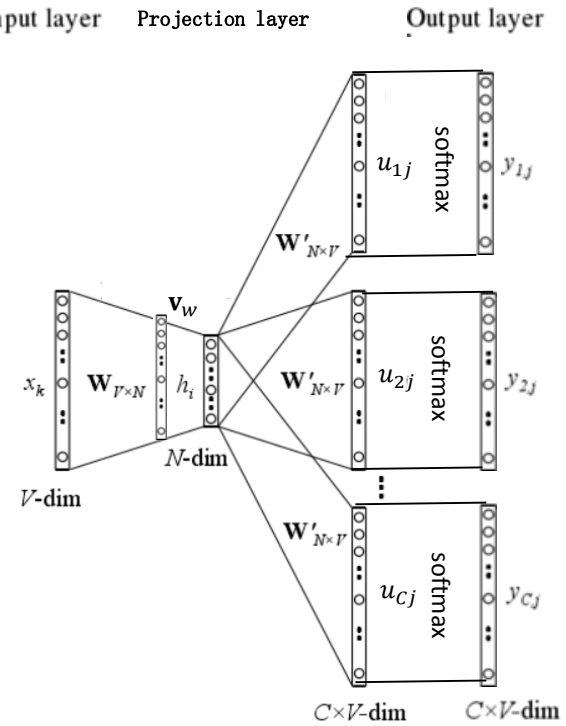
$$\mathbf{v}'^{(\text{new})}_{w_j} = \mathbf{v}'^{(\text{old})}_{w_j} - \eta \cdot e_j \cdot \mathbf{h} \qquad \text{for } j = 1, 2, \cdots, V.$$

$$\mathbf{v}^{(\text{new})}_{w_{I,c}} = \mathbf{v}^{(\text{old})}_{w_{I,c}} - \frac{1}{C} \cdot \eta \cdot \mathrm{EH} \qquad \text{for } c = 1, 2, \cdots, C.$$

算法原理

- Skip-Gram Model
  - 模型的基本思想：用当前词预测窗口长度为C内的其他词
  - 模型的优化目标：希望预测的概率 y 和真实的中心词one-hot向量 t 一致
  - 任务最终目的：将优化的参数作为词向量的输出结果

我 非常 喜欢 抓 娃娃



$$E = -\log p(w_{O,1}, w_{O,2}, \cdots, w_{O,C} | w_I)$$
$$= -\log \prod_{c=1}^{C} \frac{\exp(u_{c,j_c^*})}{\sum_{j'=1}^{V} \exp(u_{j'})}$$
$$= -\sum_{c=1}^{C} u_{j_c^*} + C \cdot \log \sum_{j'=1}^{V} \exp(u_{j'})$$

$$\mathbf{v}'^{(\text{new})}_{w_j} = \mathbf{v}'^{(\text{old})}_{w_j} - \eta \cdot \text{EI}_j \cdot \mathbf{h} \qquad \text{for } j = 1, 2, \cdots, V.$$

$$\mathbf{v}^{(\text{new})}_{w_I} = \mathbf{v}^{(\text{old})}_{w_I} - \eta \cdot \text{EH}$$

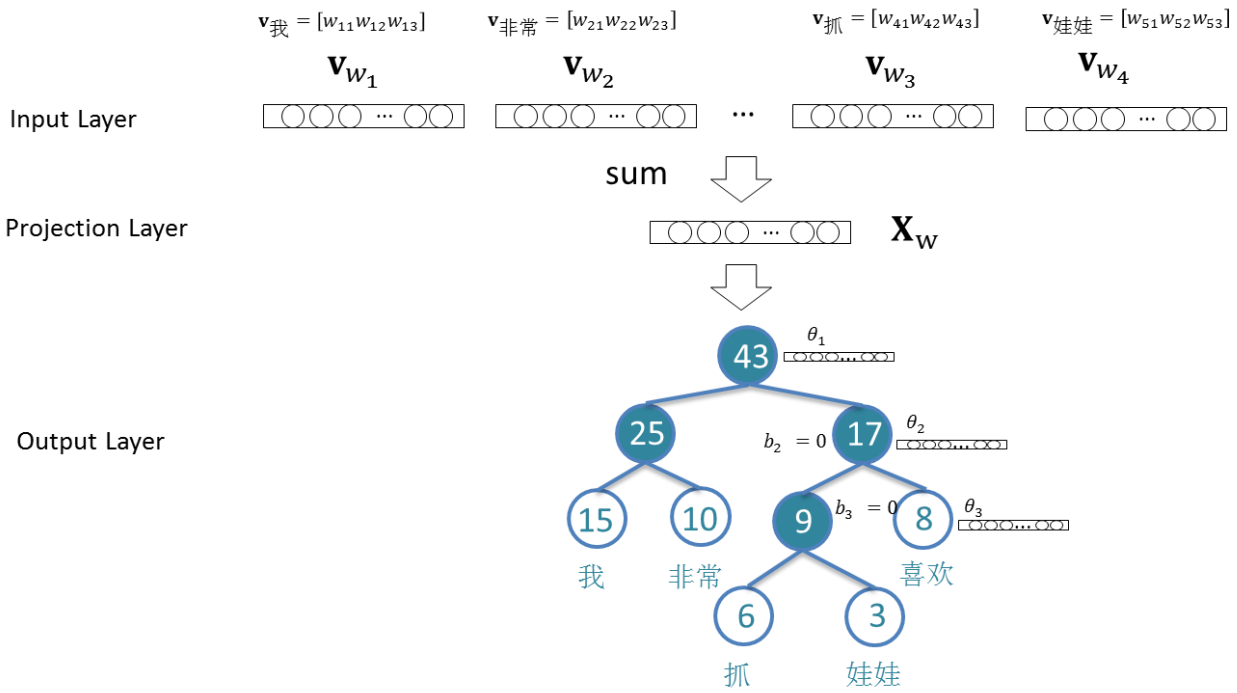
算法原理

- 两个模型存在的问题
  - 投影层到输出层的矩阵乘法运算量过大

- 改进的方法
  - Hierarchical Softmax（层级分类法）
  - Negative Sampling(负采样法)

算法原理

- 基于Hierarchical Softmax的CBOW模型
  - 模型的基本思想：用窗口中的词的向量求平均之后来预测中心词对应的路径
  - 模型的优化目标：希望真实中心词对应的路径概率最大
  - 任务最终目的：将优化的参数作为词向量的输出结果



算法原理

- 基于Hierarchical Softmax的CBOW模型

$$\mathbf{W} = \begin{matrix} \text{我} \\ \text{非常} \\ \text{喜欢} \\ \text{抓} \\ \text{娃娃} \end{matrix} \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \\ w_{51} & w_{52} & w_{53} \end{bmatrix}$$

Input Layer

$\mathbf{v}_{\text{我}} = [w_{11}w_{12}w_{13}]$ $\mathbf{v}_{\text{非常}} = [w_{21}w_{22}w_{23}]$ $\mathbf{v}_{\text{抓}} = [w_{41}w_{42}w_{43}]$ $\mathbf{v}_{\text{娃娃}} = [w_{51}w_{52}w_{53}]$

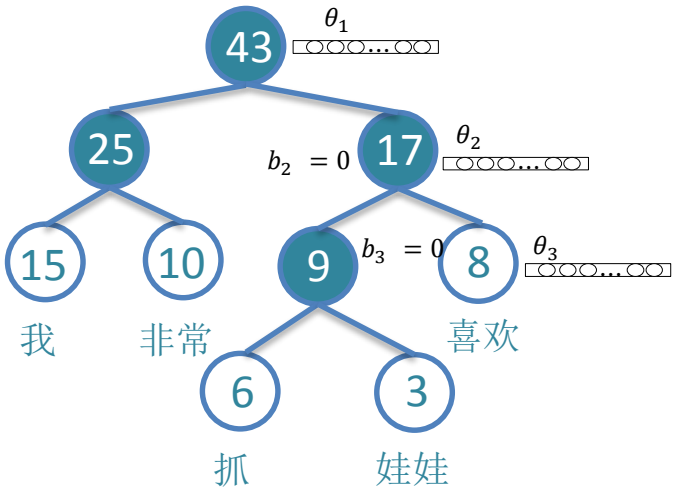
$\mathbf{v}_{w_1}$ $\mathbf{v}_{w_2}$ ... $\mathbf{v}_{w_3}$ $\mathbf{v}_{w_4}$

Projection Layer

sum

$\mathbf{X}_w$

Output Layer



$$p(w|w_m) = \prod_{i=2}^{s} p(b_i|x_w, \theta_{i-1})$$

$$\mathrm{p}(b_i|x_w, \theta_{i-1}) = \begin{cases} \sigma(x_w^T\theta_{i-1}), & b_i = 0 \\ 1 - \sigma(x_w^T\theta_{i-1}), & b_i = 1 \end{cases}$$

$$p(b_i|x_w, \theta_{i-1}) = [\sigma(x_w\theta_{i-1})]^{1-b_i} * [1 - \sigma(x_w\theta_{i-1})]^{b_i}$$

$$T = \sum_{w \in l} log(\prod_{i=2}^{s} [\sigma(x_w\theta_{i-1})]^{1-b_i} * [1 - \sigma(x_w\theta_{i-1})]^{b_i})$$

$$\frac{\partial T}{\partial \theta_i - 1} = \frac{\partial \sum_{w \in l} log(\prod_{i-2}^{s} [\sigma(x_w\theta_{i-1})]^{1-b_i} * [1-\sigma(x_w\theta_{i-1})]^{b_i})}{\partial \theta_i - 1} = [1 - b_i - \sigma(x_w\theta_{i-1})] * x_w$$

$$\frac{\partial T}{\partial x_w} = \frac{\partial \sum_{w \in l} log(\prod_{i-2}^{s} [\sigma(x_w\theta_{i-1})]^{1-b_i} * [1-\sigma(x_w\theta_{i-1})]^{b_i})}{\partial x_w} = [1 - b_i - \sigma(x_w\theta_{i-1})] * \theta_{i-1}$$

算法原理

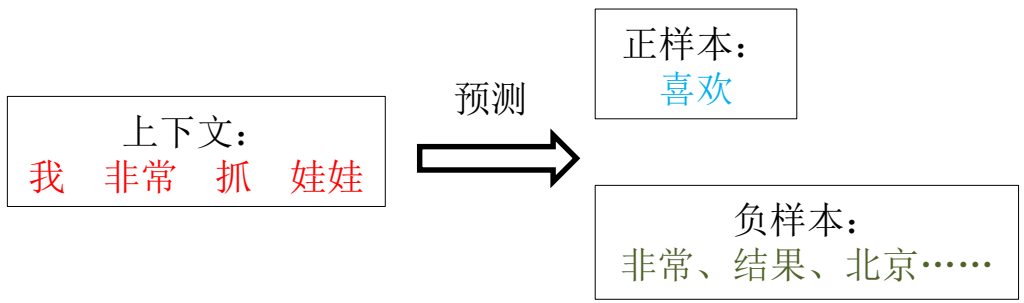
- 基于Negative Sampling的CBOW模型
  - 模型的基本思想：用窗口中的词的向量求平均之后来提高预测正样本概率的同时降低负样本的概率
  - 模型的优化目标：增大正样本概率的同时降低负样本的概率
  - 任务最终目的：将优化的参数作为词向量的输出结果

我 非常 喜欢 抓 娃娃

优化的目标函数：$g(w) = \prod_{u \epsilon \{w\} \cup N} EG(w) \; p(u|Context(w))$

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

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



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

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

特点分析

特点分析

- Word2vec流行的原因主要由以下三点
  - 极快的训练速度
  - 一个酷炫的man-woman=king-queen的示例
  - Word2vec里有大量的tricks

VS

〃 〃

语言模型 VS word2vec

应用总结

应用总结

- 同义词挖掘
- 构建句子向量
- 生成其他序列数据的向量
- 作为另一个模型的输入
  - 机器翻译
  - 文本摘要
  - 情感分析
  - ……

参考资料

[1] http://blog.csdn.net/mytestmy/article/details/26969149 深度学习word2vec笔记之算法篇

[2] http://blog.csdn.net/itplus/article/details/37969979 word2vec 中的数学原理详解

[3] http://xiaoquanzi.net/?p=156 hisen博客的博文

[4] Hierarchical probabilistic neural network language model. Frederic Morin and Yoshua Bengio.

[5] Distributed Representations of Words and Phrases and their Compositionality T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean.

[6] A neural probabilistic language model Y. Bengio, R. Ducharme, P. Vincent.

[7] Linguistic Regularities in Continuous Space Word Representations. Tomas Mikolov,Wen-tau Yih,Geoffrey Zweig.

[8] Efficient Estimation of Word Representations in Vector Space. Tomas Mikolov,Kai Chen,Greg Corrado,Jeffrey Dean.

[9]http://licstar.net/archives/328 Deep Learning in NLP （一）词向量和语言模型

道德经

知人者智，自知者明。

胜人者有力，自胜者强。

知足者富。

强行者有志。

不失其所者久。

死而不亡者，寿。

谢谢！