预训练词向量

常宝宝
北京大学计算语言学研究所
chbb@pku.edu.cn

概要

- **词向量概要**
- 预测式词向量学习模型
  - Collobert&Weston 模型
  - CBOW模型/SkipGram模型
- 矩阵分解式词向量学习模型
  - SVD分解模型
  - GloVe模型
- 词向量评价
  - 类比词预测
  - 相似度评价

词的表示

- 词的one-hot向量表示

  令V代表词表，$w_i \in V$可表示成一个$|V|$维向量，词表中每个词单独对应一个向量维度。

$$v(w_i) = \begin{matrix}(\ 0 & 0 & \cdots & 1 & \cdots & 0\ )\\ 1 & 2 & \cdots & i & \cdots & |V|\end{matrix}$$

  唯有与w_i对应的维度为1，其余维度为0

- 特点：
  - 稀疏、高维
  - 正交：若$w_i, w_j \in V$且$i \neq j$

$$v(w_i) \cdot v(w_j) = 0$$

- 不能有效表示词和词之间的句法语义共性

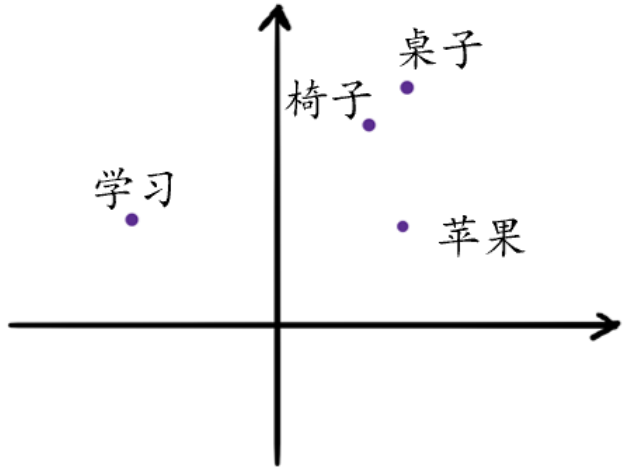
词向量

- one-hot表示→分布式表示
- 词的嵌入表示——词向量(word vector/embedding)
- 把词表示成低维空间中的向量

- 句法语义特性相近的词在空间中距离相近

$$v(\text{桌子}) \cdot v(\text{椅子}) \neq 0$$

- 以词向量承载句法语义信息



词向量的习得

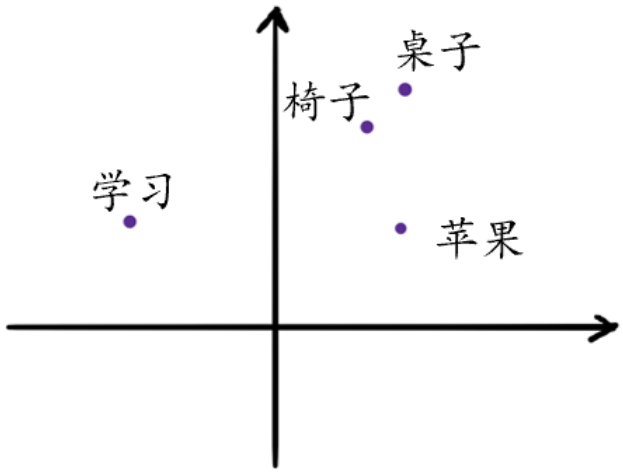
- 词义相似性 常常 反映为分布相似性
- 分布假设(Distributional Hypothesis)

  Words that occur in similar contexts tend to have similar meanings.

- 根据词的分布规律学习词的表示

- 基于词的分布特性自动习得词向量
  - 大规模未标注语料库

词的表示



- 不同的词是不同的符号标识
  - 桌子、椅子、苹果、学习
  - 无异于四个不同的编号
- 词和词在语义或语法方面的共性
  - 桌子、椅子都是家具、苹果是水果、学习是一种行为
- 分布相似性
  - 木头桌子(√)、木头椅子(√) 、木头苹果(×)、木头学习(×)
  - 一把椅子(√) 、一张桌子(√)、一个苹果(√)、一个学习(×)
  - ……

概要

- 词向量概要
- **预测式词向量学习模型**
  - Collobert&Weston 模型
  - CBOW模型/SkipGram模型
- 矩阵分解式词向量学习模型
  - SVD分解模型
  - GloVe模型
- 词向量评价
  - 类比词预测
  - 相似度评价

预测式模型

- 给定上下文语境预测目标词

$$p(w_t|w_{t-l}w_{t-l+1}\dots w_{t-1}w_{t+1}\dots w_{t+l})$$

- 语境和目标词
  - 左侧语境 $w_{t-l}w_{t-l+1}\cdots w_{t-1}$
  - 右侧语境 $w_{t+1}w_{t+2}\cdots w_{t+l}$
  - 语境窗口宽度l

- 寻求如下函数

$$p(w_t|w_{t-l}\cdots w_{t-1}w_{t+1}\cdots w_{t+l})$$
$$= f\left(v(w_{t-l}),\cdots,v(w_{t-1}),v(w_{t+1}),\cdots,v(w_{t+l})\right)$$

填空

他 坐 在 _____ 上 看 书

椅子？ 桌子？ 苹果？ 学习？

'坐'的上下文中可以有'椅子',
'坐'的向量表示应该体现出这一点!

预测式模型

- 训练数据 $Text = w_1 w_2 \ldots w_T$
- 利用未标注语料按照窗口宽度提取训练例子 $(bw_2, w_1), (w_1 w_3, w_2), \cdots, (w_{T-2} w_T, w_{T-1}), (w_{T-1} e, w_T)$
- 目标函数 –平均对数似然函数

$$J_\theta = \frac{1}{T} \sum_{t=1}^{T} \log p(w_t | w_{t-l} \cdots w_{t-1} w_{t+1} \cdots w_{t+l}; \theta)$$

- 最大似然估计：寻求能使目标函数值最大化的词向量赋值方案

$$\hat{\theta} = \text{argmax}_\theta J_\theta$$

$$\theta = \left(v(w_1), v(w_2), \ldots, v\left(w_{|V|}\right) \right)$$

神经网络建模

- 输入层

$$x = [v(w_{t-l}); \cdots; v(w_{t-1}); v(w_{t+1}); \cdots; v(w_{t+l})]$$

- 非线性变换层

$$h = g(W^{(1)}x + b^{(1)})$$

- 线性变换

$$o = W^{(2)}h + b^{(2)}$$

思考$W^{(2)}$的维度

- softmax层

$$p(w_t|w_{t-l}, \cdots, w_{t-1}w_{t+1}, \cdots, w_{t+l}) = \frac{\exp(o_{w_t})}{\sum_{w \in V} \exp(o_w)}$$

神经网络建模

- 计算代价昂贵，无法针对大词表生成词向量
- 目标词预测模型 ⇒ 语言片段打分模型
- 学习区分正确的语言片段和错误的语言片段
- 负例生成(将目标词w_t随机替换为w_t')

$$w_{t-l}\cdots w_{t-1}w_t w_{t+1}\cdots w_{t+l} \Rightarrow w_{t-l}\cdots w_{t-1}w_t' w_{t+1}\cdots w_{t+l}$$

- 给定$Text = w_1 w_2 \cdots w_T$，提取训练数据

$$\begin{aligned}
&(bw_1w_2, && bw_1'w_2)\\
&(w_1w_2w_3, && w_1w_2'w_3)\\
&\quad\vdots && \quad\vdots\\
&(w_{T-2}w_{T-1}w_T, && w_{T-2}w_{T-1}'w_T)\\
&(w_{T-1}w_Te, && w_{T-1}w_T'e)
\end{aligned}$$

判断

他 坐 在 **椅子** 上 看 书 (√)
他 坐 在 **苹果** 上 看 书 (×)

Score(他 坐 在 **椅子** 上 看 书) > Score(他 坐 在 **苹果** 上 看 书)

神经网络建模

- 输入层

$$x = [v(w_{t-l}); \cdots; v(w_{t-1}); v(w_t); v(w_{t+1}); \cdots; v(w_{t+l})]$$

- 非线性变换层

$$h = g(W^{(1)}x + b^{(1)})$$

- 线性变换

$$f_\theta(w_{t-l} \cdots w_{t-1} w_t w_{t+1} \cdots w_{t+l}) = w^\top h + b$$

$w^\top$是向量

- 排序损失(pairwise rank loss)

$$J_\theta = \sum_{s \in S} \sum_{w \in V} \max(0, 1 - f(s) + f(s^w))$$

s正确语言片段, s^w是把s中目标词随机替换为w得到的语言片段

正确语言片段得分高于错误语言片段，二者间隔至少是1

Collobert & Weston模型

- 语言片段长度为11，即$l = 5$
- 非线性激活函数HardTanh
- $|V| = 30000$
- 隐层维度100
- 词向量维度50
- 训练语料：Wikipedia (631M words)

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED
454	1973	6909	11724	29869
SPAIN	CHRIST	PLAYSTATION	YELLOWISH	SMASHED
ITALY	GOD	DREAMCAST	GREENISH	RIPPED
RUSSIA	RESURRECTION	PSNUMBER	BROWNISH	BRUSHED
POLAND	PRAYER	SNES	BLUISH	HURLED
ENGLAND	YAHWEH	WII	CREAMY	GRABBED
DENMARK	JOSEPHUS	NES	WHITISH	TOSSED
GERMANY	MOSES	NINTENDO	BLACKISH	SQUEEZED
PORTUGAL	SIN	GAMECUBE	SILVERY	BLASTED
SWEDEN	HEAVEN	PSP	GREYISH	TANGLED
AUSTRIA	SALVATION	AMIGA	PALER	SLASHED

CBOW模型(continuous bag-of-words)

- 给定上下文语境预测目标词

$$p(w_t|w_{t-l}w_{t-l+1}\cdots w_{t-1}w_{t+1}\cdots w_{t+l})$$

- 输入层

$$x=\sum_{w\in C(w_t)} vec(w)$$

- 输出层(取消隐层)

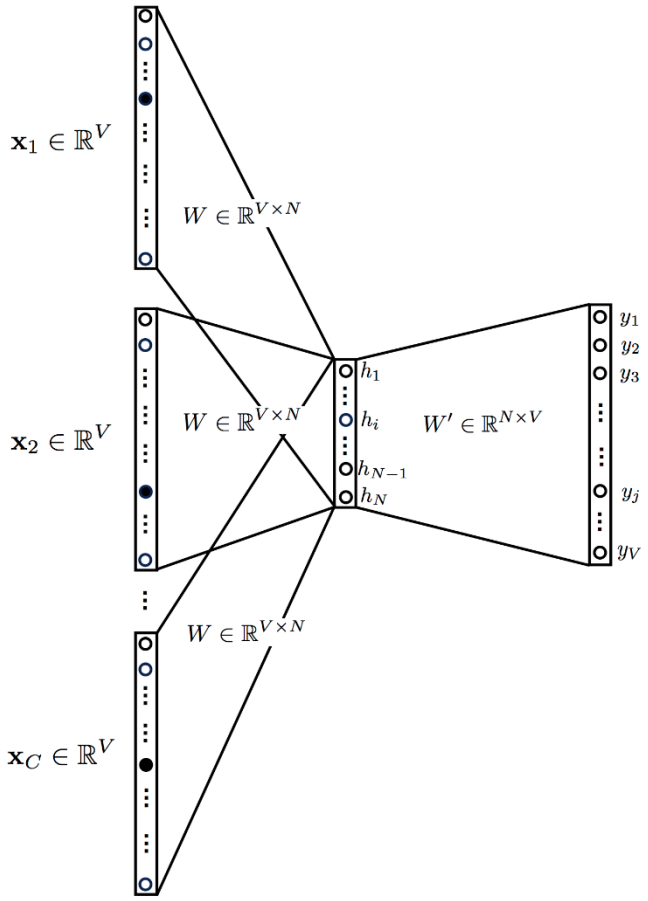
$$o=Ux, U\in \mathbb{R}^{|V|\times d}$$

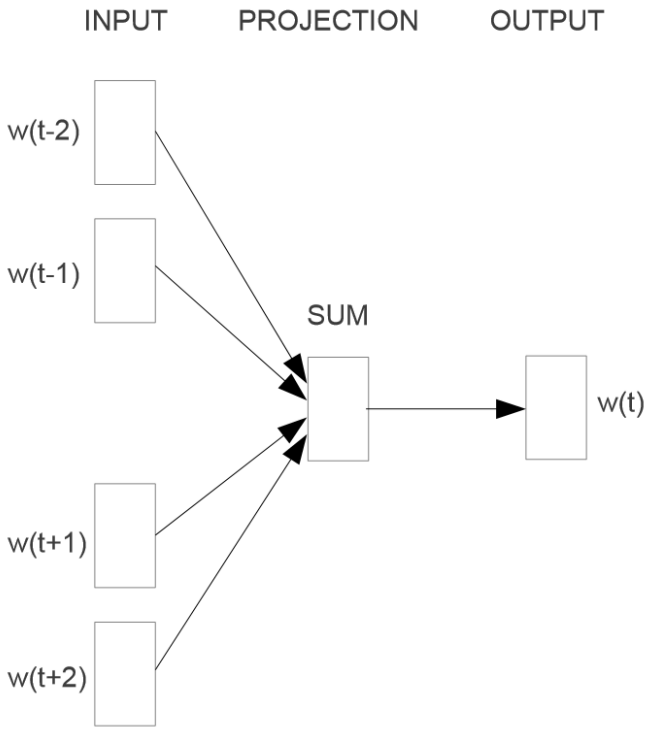
- Softmax层

$$p(w_t|w_{t-l},\cdots,w_{t-1}w_{t+1},\cdots,w_{t+l})=\frac{\exp(o_{w_t})}{\sum_{w\in V}\exp(o_w)}$$

CBOW模型

- 语境词序无影响，词袋模型





预测式模型

- 给定目标词预测语境中的词

$$p(w_{t+i}|w_t)$$
$$i = -l, \cdots, -1, +1, \cdots, +l$$

- 训练数据 $Text = w_1 w_2 \ldots w_T$
- 利用未标注语料按照窗口宽度提取训练例子
  $(w_1, w_2), (w_2, w_1), (w_2, w_3) \cdots, (w_{T-1}, w_T), (w_T, w_{T-1})$
- 目标函数 –平均对数似然函数

$$J_\theta = \frac{1}{T} \sum_{t=1}^{T} \sum_{-l \le i \le l, i \neq 0} \log p(w_{t+i}|w_t; \theta)$$

填空

__ __ __ **椅子** __ __ __

↓ ↓ ↓ ↓ ↓ ↓

他 坐 在 上 看 书

SkipGram

- 输入层

$$x = vec(w_t) = v_{w_t}$$

- 输出层(线性变换)

$$o = Uv_w, U \in \mathbb{R}^{|V| \times d}$$

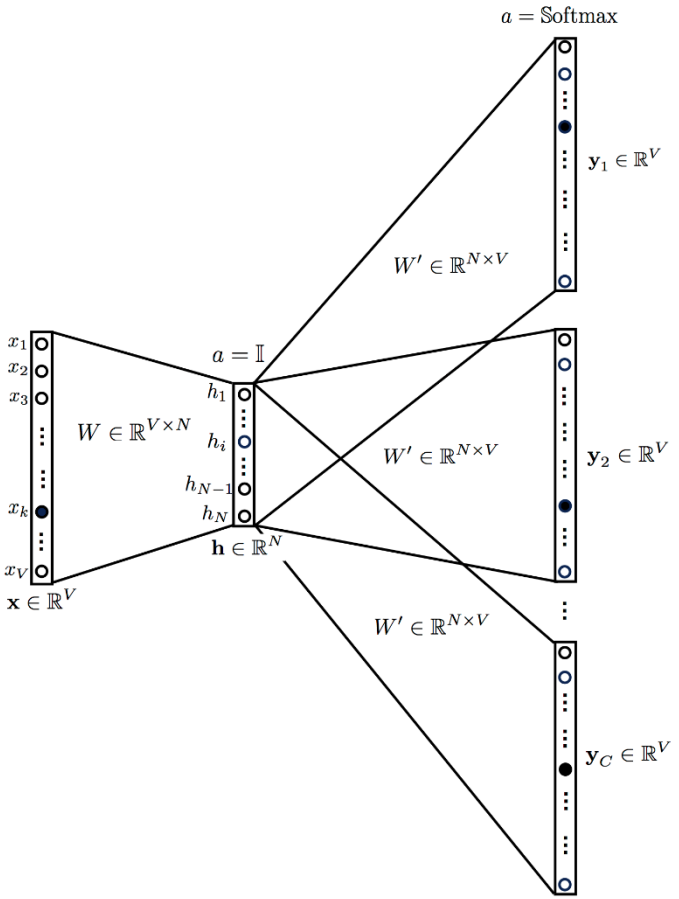
- Softmax层

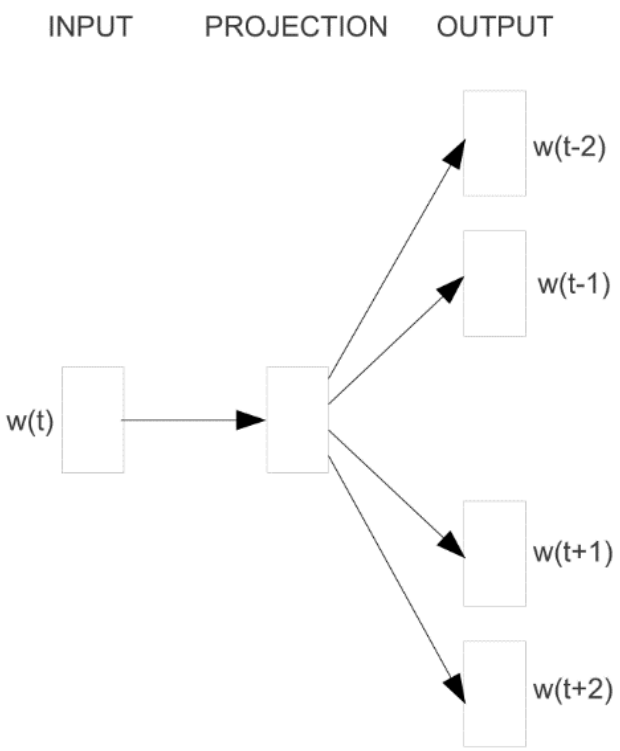
$$p(w_c|w_t) = \frac{\exp(o_{w_c})}{\sum_{w \in V} \exp(o_w)}$$

$$o_{w_c} = v_{w_c} \cdot v_{w_t}$$

- v_{w_c}是矩阵U中词w_c对应的行（输出层词向量）

SkipGram


$a = \text{Softmax}$
$\mathbf{y}_1 \in \mathbb{R}^V$
$W' \in \mathbb{R}^{N \times V}$
$a = \mathbb{I}$
x_1
x_2
x_3
$W \in \mathbb{R}^{V \times N}$
h_1
h_i
h_{N-1}
h_N
x_k
x_V
$\mathbf{x} \in \mathbb{R}^V$
$\mathbf{h} \in \mathbb{R}^N$
$W' \in \mathbb{R}^{N \times V}$
$\mathbf{y}_2 \in \mathbb{R}^V$
$W' \in \mathbb{R}^{N \times V}$
$\mathbf{y}_C \in \mathbb{R}^V$


INPUT
PROJECTION
OUTPUT
w(t)
w(t-2)
w(t-1)
w(t+1)
w(t+2)

负采样训练

- 基于整个词表归一, softmax归一代价大
- Negative sampling
  - 给定目标词w
  - 源自训练数据的样本(w, c)正例
  - 随机生成负样本(w, c') 负例
- 二分类：(w, c)是否源自训练数据

  $p(D = 1|w, c; \theta)$ 代表 (w, c)源自训练数据的概率

  $p(D = 0|w, c; \theta)$代表 (w, c) 不是源自训练数据的概率

$$p(D = 0|w, c; \theta) = 1 - p(D = 1|w, c; \theta)$$

负采样训练

$$p(D=1|w,c;\theta)=\frac{1}{1+\exp(-v_w\cdot v_c)}=\sigma(v_w\cdot v_c)$$
$$p(D=0|w,c;\theta)=1-p(D=1|w,c;\theta)=\sigma(-v_w\cdot v_c)$$

- 对数似然函数/最大似然估计

$$J_\theta=\log\prod_{(w,c)\in D}p(D=1|w,c;\theta)\prod_{(w,c)\in D'}p(D=0|w,c;\theta)$$
$$=\sum_{(w,c)\in D}\log p(D=1|w,c;\theta)+\sum_{(w,c)\in D'}\log p(D=0|w,c;\theta)$$
$$=\sum_{(w,c)\in D}\log\sigma(v_w\cdot v_c)+\sum_{(w,c)\in D'}\log\sigma(-v_w\cdot v_c)$$

$$\hat{\theta}=\operatorname*{argmax}_\theta J_\theta=\operatorname*{argmax}_\theta\left(\sum_{(w,c)\in D}\log\sigma(v_w\cdot v_c)+\sum_{(w,c)\in D'}\log\sigma(-v_w\cdot v_c)\right)$$

负例构造方式

- 负例生成

  对任意$(w, c) \in D$，按如下分布随机生成c_i

$$c_i \sim \frac{1}{Z} U(w)^{\frac{3}{4}}, i = 1,2, \dots, k$$

  令

$$(w, c_i) \in D'$$

- 语境窗口的确定
  - 动态确定，设定最大窗口宽度l, 针对每个目标词w,随机确定窗口宽度$l' \sim [1, l]$
  - 排除低频词，对高频词用例做降采样

概要

- 词向量概要
- 预测式词向量学习模型
  - Collobert&Weston 模型
  - CBOW模型/SkipGram模型
- **矩阵分解式词向量学习模型**
  - SVD分解模型
  - GloVe模型
- 词向量评价
  - 类比词预测
  - 相似度评价

矩阵分解式模型

- 目标词w及其语境中出现的词c

$$w, c \in V, (w, c) \in D$$

- 定义w和c的共现矩阵M
  - 行对应目标词w
  - 列对应语境词c
  - M_{ij}代表w_i和c_j的某种关联度
- 最简单的关联度是w和c的共现次数

$$M_{ij} = \#(w_i, c_j)$$

- 原始共现频次的缺陷
  - 分布方差巨大，高频词(如虚词)影响高估

共现矩阵M

上下文语境中的词

目标词	c_1	c_2	…	c_j=坐	…	$c_{\|V\|}$
w_1	*	*	…	*	…	*
w_2	*	*	…	*	…	*
…	…	…	…	…	…	…
w_i=椅子	*	*	…	78	…	*
…	…	*	…	*	…	…
$w_{\|V\|}$	*	*	…	*	…	*

PMI和PPMI

- 点间互信息(PMI)

$$PMI(w,c) = \log \frac{\hat{p}(w,c)}{\hat{p}(w) \cdot \hat{p}(c)} = \log \frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)}$$

- 若$\#(w,c) = 0$,则$PMI(w,c) = -\infty$
- 正点间互信息(PPMI)

$$PPMI(w,c) = \max(PMI(w,c), 0)$$

- 令

$$M_{ij} = PPMI(w_i, c_j)$$

矩阵分解

- 奇异值分解(SVD)

$$M = U \cdot \Sigma \cdot V^{\top}$$

U和V的列向量均为单位正交向量

Σ为奇异值降序排列的对角矩阵

- 通过奇异值截断，寻求矩阵的低秩逼近矩阵

$$M_d = U_d \cdot \Sigma_d \cdot V_d^{\top}$$

保留Σ中d个最大的奇异值

矩阵分解

- 基于SVD的非对称词向量

$$W^{SVD} = U_d \cdot \Sigma_d$$
$$C^{SVD} = V_d$$

  矩阵的行对应w和c的向量表示

- 基于SVD的对称词向量

$$W^{SVD} = U_d \cdot \sqrt{\Sigma_d}$$
$$C^{SVD} = V_d \cdot \sqrt{\Sigma_d}$$

矩阵分解动机

- 词w_i 的表示取决于同其语境中词w_k的共现关系
- 寻求如下的矩阵分解

$$W \cdot C^{\top} \approx M$$

- 分解使得词向量压缩表示了词的语境信息，反映了词的句法语义信息
- 两个词的语境接近，两个词就含有接近的句法语义信息

GloVe

- w和c的关联度定义

$$M_{ij} = \log\left(\#(w_i, c_j)\right)$$

- 寻求如下的矩阵分解

$$M \approx W \cdot C^{\top} + \vec{b}_w I^{\top} + I\vec{b}_c^{\top}$$

即

$$\vec{w} \cdot \vec{c} + b_w + b_c \approx \log(\#(w, c)) \ \forall (w, c) \in D$$

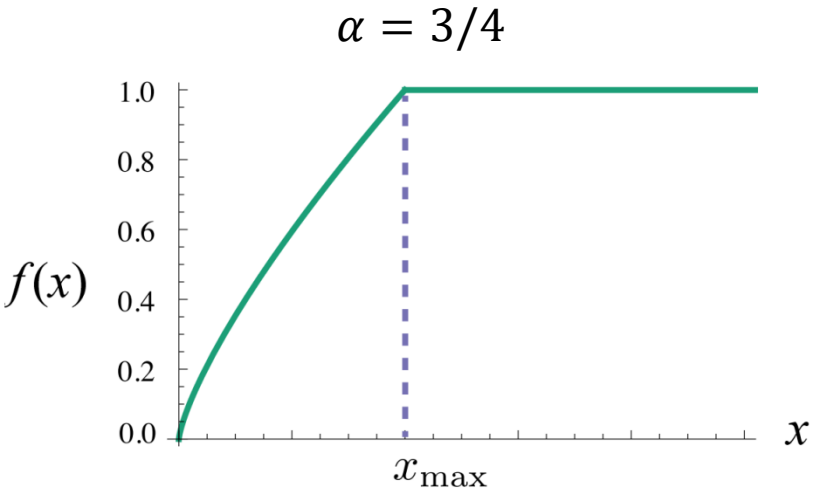
- 分解误差

$$J = \sum_{i,j} f\left(\#(w_i, c_j)\right)\left(\vec{w}_i \cdot \vec{c}_j + b_{w_i} + b_{c_j} - M_{ij}\right)^2$$

GloVe

- 权重函数(weighting function)

$$f(x) = \begin{cases} (x/x_{max})^{\alpha}, & if\ x < x_{max} \\ 1, & otherwise. \end{cases}$$



- Glove的优化目标

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} J_{\theta}$$

概要

- 词向量概要
- 预测式词向量学习模型
  - Collobert&Weston 模型
  - CBOW模型/SkipGram模型
- 矩阵分解式词向量学习模型
  - SVD分解模型
  - GloVe模型
- **词向量评价**
  - 类比词预测
  - 相似度评价

词向量的评价-word analogy task

- 预测具有同样类比关系的词，计算准确率。

  a is to *b* as *c* is to _____?

  例子：

  Athens is to *Greece* as *Berlin* is to _____? (Germany)

  dance is to *dancing* as *fly* is to _____? (flying)

  可否通过词向量计算找到具有同样类比关系的词？

  把词和词之间的关系表达为向量差，则应有

$$\vec{w}_a - \vec{w}_b \approx \vec{w}_c - \vec{x}$$

$$\vec{x} \approx \vec{w}_b - \vec{w}_a + \vec{w}_c$$

$$\hat{d} = \underset{d \in V \setminus \{a,b,c\}}{\text{argmax}} \; similarity(\vec{w}_d, \vec{x})$$

词向量的评价-word analogy task

- Mikolov et al. 2013发布词类比关系数据集
  - 19,544个类比问题
    - 8869语义类比(5类)
    - 10675句法类比(9类)

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

词向量评价-word similarity task

- 计算两个词的语义相似度(相关度)
  $similarity(happy, cheerful) =?$ (9.55)
- 依据相似度排序，计算与标准(人工)排序的相关性

  Spearman's correlation coefficient ρ

$$\rho = \frac{cov(rA, rB)}{\sigma_{rA}\sigma_{rB}}$$

$|\rho| \leq 1$ 越接近1，单调性越强，排序具有越强的相关性

.00-.19 "very weak"
.20-.39 "weak"
.40-.59 "moderate"
.60-.79 "strong"
.80-1.0 "very strong"

词向量评价-word similarity task

- wordSim-353数据集
  - 353个词对及人工相似度(相关度)评分，例如

  tiger cat 7.35

  computer keyboard 7.62

  monk oracle 5

  noon string 0.54

- SimLex-999数据集
  - 999个词对及人工相似度评分，例如
  - happy cheerful 9.55
  - bad terrible 7.78
  - old new 1.58
  - disc computer 3.2