Lecture 2：Word2Vec

2018年7月4日 星期三　　上午9:16

1.Word meaning

Signifier -> Signified idea = denotation

? Context words是否真的能够描述center words的语义？ 普通人说话的上下文一般不会再次解释形容词的含义。
是否可以从词典的释义中提取一定数目的词作为context words?

NLTK WordNet
Weakness of Discrete representation:

- Synonyms missing nuance
- Missing new words
- Require human labor
- one-hot coding(Symbolic)
  - Very large dimension
  - No relationship between words(dot product is 0)

Distributional similarity
Get a lot of neighbor words to represent its meaning
Distributed Representation
Build a dense vector for each word

2.Word2Vec (Mikolov 2013)

Simple, fast to train, Scalable
E.g. Cat-kitty+dog ~ puppy
通过大量文本以无监督方式学习词向量，在嵌入空间中表征语义信息。

? 多义词的向量仅有一个，语义是否会造成模糊

- **Two Algorithms**
  Fake Task，真正需要的是获得模型参数（Word Vectors）
  - 1.SG (Skip-grams)
    - 给定center word 预测 context
    - $L = \sum_{w \in C} \log \mathrm{P}(\mathrm{content(w)|w})$
  - 2.CBOW (Continuous Bag of Words)
    - 给定context 预测 center word
    - $L = \sum_{w \in C} \log \mathrm{P}(\mathrm{w|content(w)})$
- **Two Training Methods**
  Motivation：softmax over all data need O(N) time->Impractical
  - 1.Hierarchical Softmax
• Huffman encoding:	reduce parameters number
• Product along the path:	Reduce compute time to O(log(n))
  - 2.Negative Sampling
    - Sample some negative cases rather than whole data
      Reduce softmax summation number

3.Softmax :

- Standard map from R^V into probability distribution.
- Using center word C to obtain probability of word O

- $P_i = \frac{e^{u_i}}{\sum_j e^{u_j}}$

$$\frac{\partial P_i}{\partial u_j} = \begin{cases} P_i(1-P_i) & i=j \\ -P_i P_j & i \neq j \end{cases}$$

In Word2Vec :

$$P(O|C) = \frac{\exp(u_o^T \cdot v_c)}{\sum_{w=1}^{V} \exp(u_w^T \cdot v_c)}$$

<u, v> : loose measure of simmilarity

4.Skipgram

- A bag-of-words model:
  - **not** care about words' **position**
- Target :
  - Maximize the product of probability of neighbors
  - Obtain intermediate parameters(word vectors)

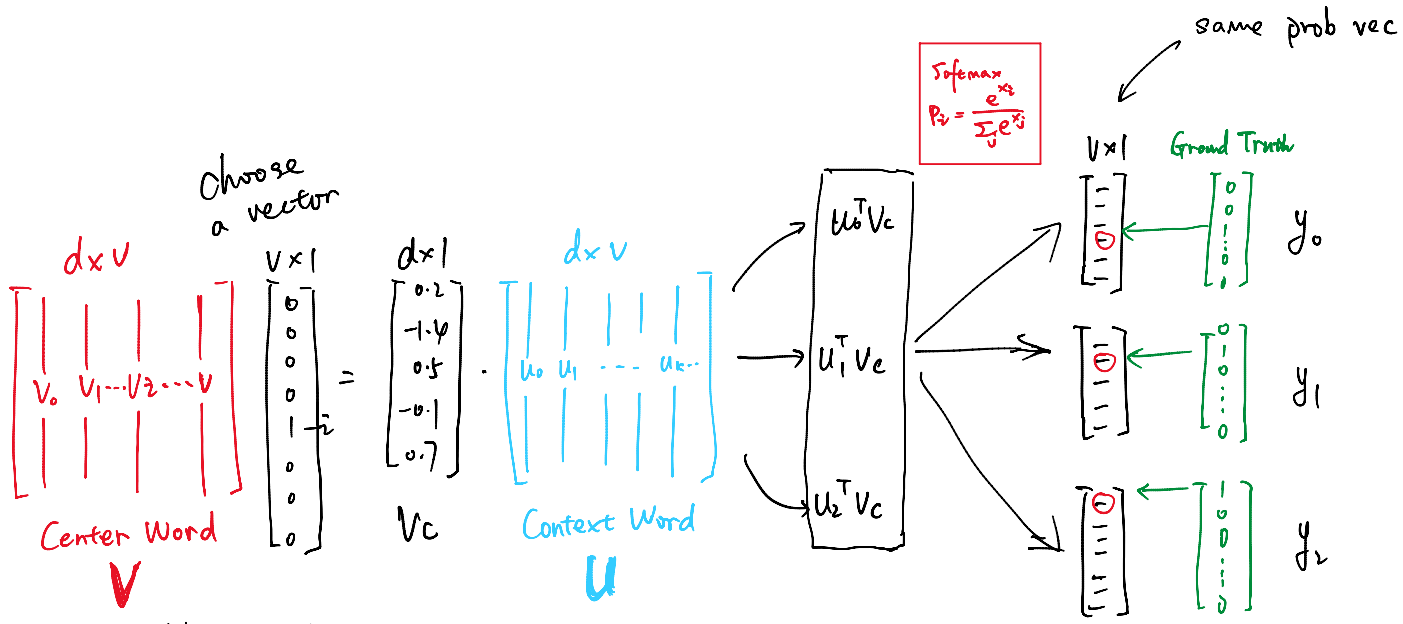
Object function T: text length , m: window size.

$$J(\theta) = \prod_{t=1}^{T} \prod_{-m \le j \le m} P(w_{t+j}|w_t;\theta)$$

Negative Log Likelihood

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

? 使用对数的原因是求导方便，那为何要加负号?



Parameter:

$$\theta = [V \mid U] = [v_0 v_1 \cdots v_V \mid u_0 u_1 \cdots u_V]$$

5.Training with Negative Sampling:

Actually in a 3-layer Neural Network

minimize: $J(\theta) = -\frac{1}{T}\sum_t \sum_j \log(p(w_{t+j}|w_c))$

Global?

$$\frac{\partial}{\partial v_c} \log \frac{e^{u_o^T v_c}}{\sum_w \exp(u_w^T v_c)} = \frac{\partial}{\partial v_c}\left(u_o^T v_c - \log \sum_w \exp(u_w^T v_c)\right)$$

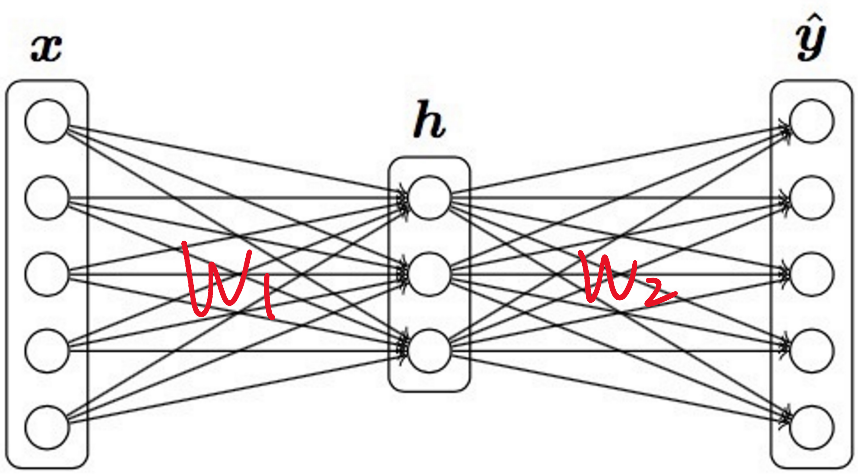
$$\frac{\partial}{\partial v_c}\log\frac{e^{u_o^T v_c}}{\sum_w \exp(u_w^T v_c)} = \frac{\partial}{\partial v_c}\left(u_o^T v_c - \log\sum_w \exp(u_w v_c)\right)$$

$$= u_o - \frac{\sum_x \exp(u_x^T \cdot v_c)\cdot u_x}{\sum_w \exp(u_w^T v_c)}$$

$$= u_o - \sum_x \cdot \frac{\exp(u_x^T v_c)}{\sum_w \exp(u_w^T v_c)} \cdot u_x$$

$$= u_o - \sum_x P(x|c)\cdot u_x \qquad u_o - U\cdot\sigma(U^T V)[:,c]$$

= Observed - Expected.



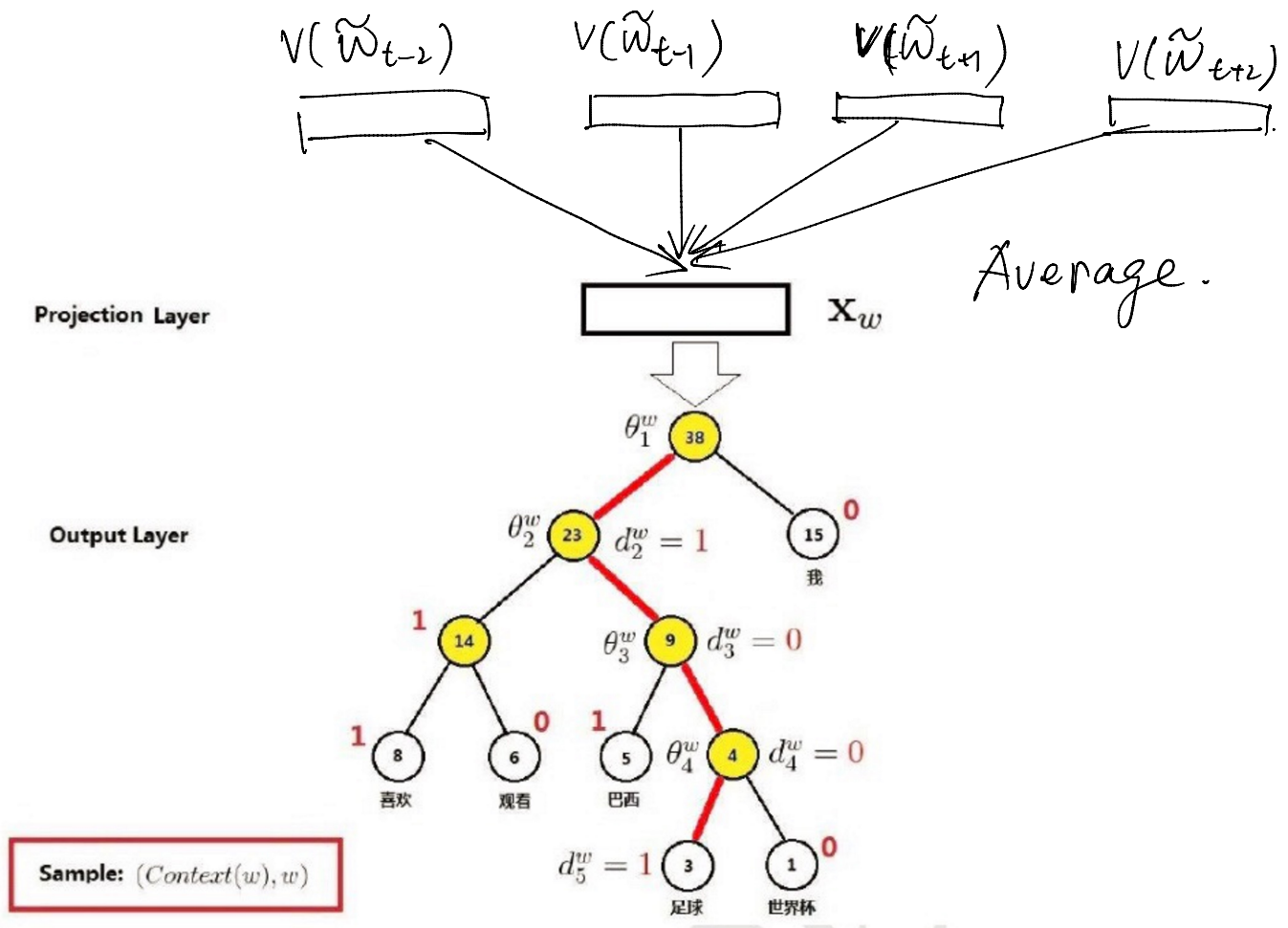
W1： Center Word vectors

W2: Context Word vectors

The final word vectors are W1 concat W2

6.Training with Hierarchical Softmax：

https://www.cnblogs.com/Determined22/p/5807362.html



直接取平均是否不妥，按距离加权？

现实中Negative Sampling与Hierarchical Softmax哪一个表现更好呢？

对于：

$$p(d_4^w \mid X_w, \theta_3^w) = \sigma(X_w^T \theta_3^w)$$

$$p(w_4 \mid x_w, \theta_3) = \sigma(x_w^T \theta_3)$$

$$\mathcal{L} = p(\text{足球} \mid \text{context}(\text{足球})) = \prod_{j \sim path} p(d_j^w \mid x_w, \theta_{j-1}^w)$$

更新：

$$v(\tilde{w}) := v(\tilde{w}) + \eta \cdot \frac{\partial \mathcal{L}(w, j)}{\partial x_w}, \quad \tilde{w} \in context(w)$$

$$\theta_{j-1} := \theta_{j-1} + \eta \cdot \frac{\partial \mathcal{L}(w, j)}{\partial \theta_{j-1}}$$