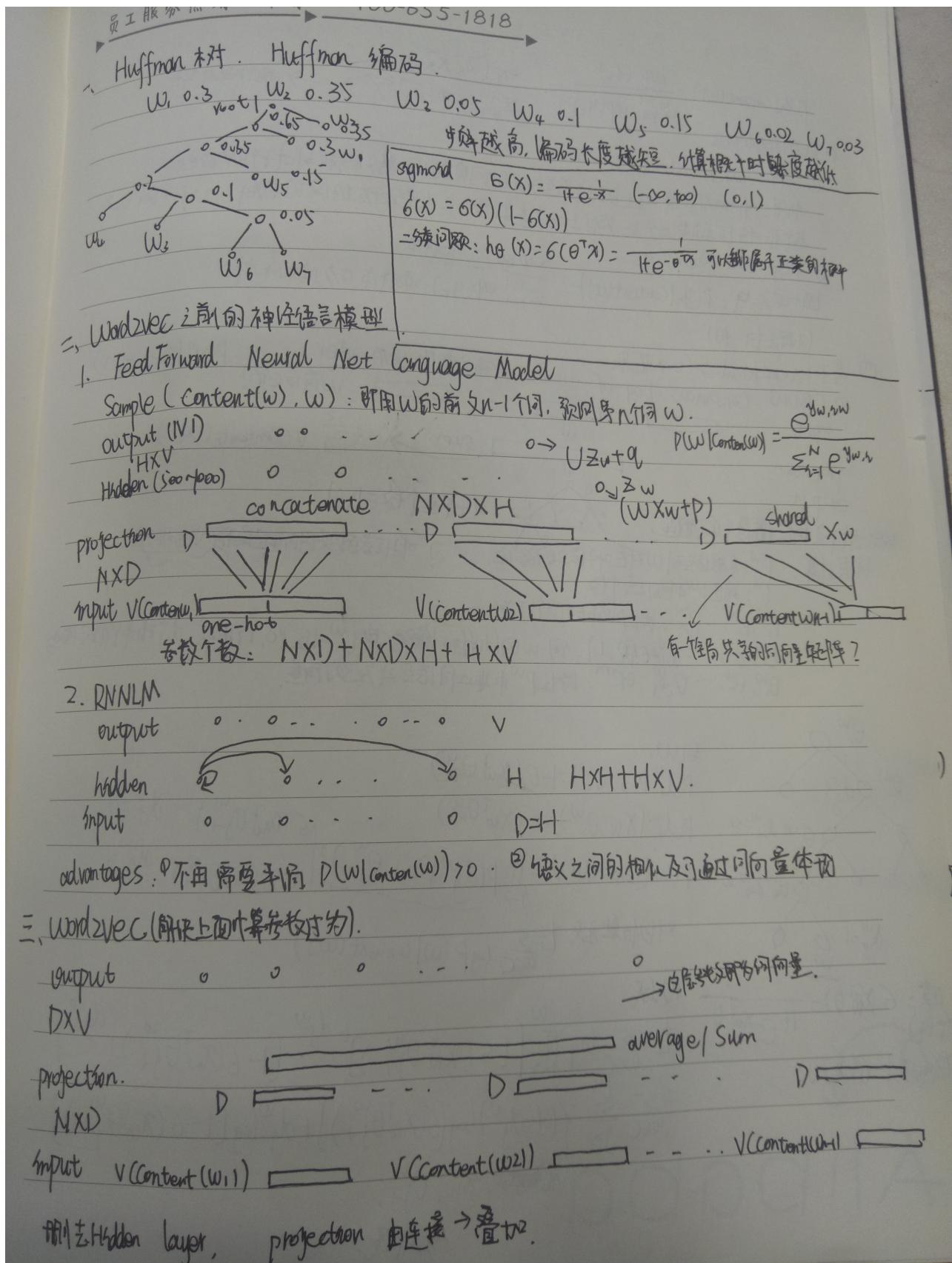


转载自 <https://www.zybuluo.com/DouNm/note/591752>

<http://www.cnblogs.com/peghoty/p/3857839.html>



$$P(y_n | \text{Content}(w)) = \frac{\exp(y_n)}{\sum_{k=1}^{|V|} \exp(y_k)} = \frac{\exp(\vec{w}_n^\top \vec{x})}{\sum_{k=1}^{|V|} \exp(\vec{w}_k^\top \vec{x})} \rightarrow \text{对每个词的得分}$$

优化的两种方法

① Hierarchical Softmax

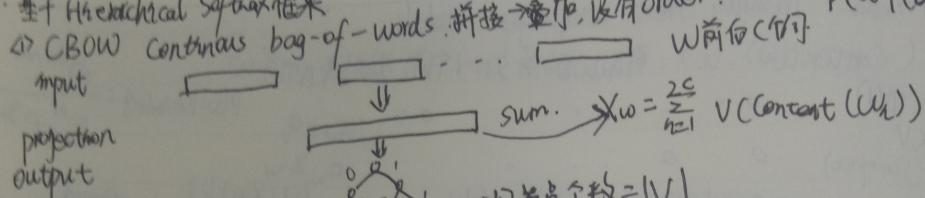
利用 Huffman 树，将树上的叶子节点分配给词典中的词，将从树根到叶节点的路径上所有非叶子节点或二分类路线上与类连结的不是叶节点对应的词的权值。 $w \rightarrow \log w$

② Negative Sampling.

随机负采样。 $P(y_n | \text{Content}(w)) = \frac{\exp(y_n)}{\sum_{k=1}^{|V|} \exp(y_k)} \rightarrow$ 负样本，多，随机采样。

(不需归一化)。

④ 基于 Hierarchical Softmax 框架



输出层和投影层直接相连。

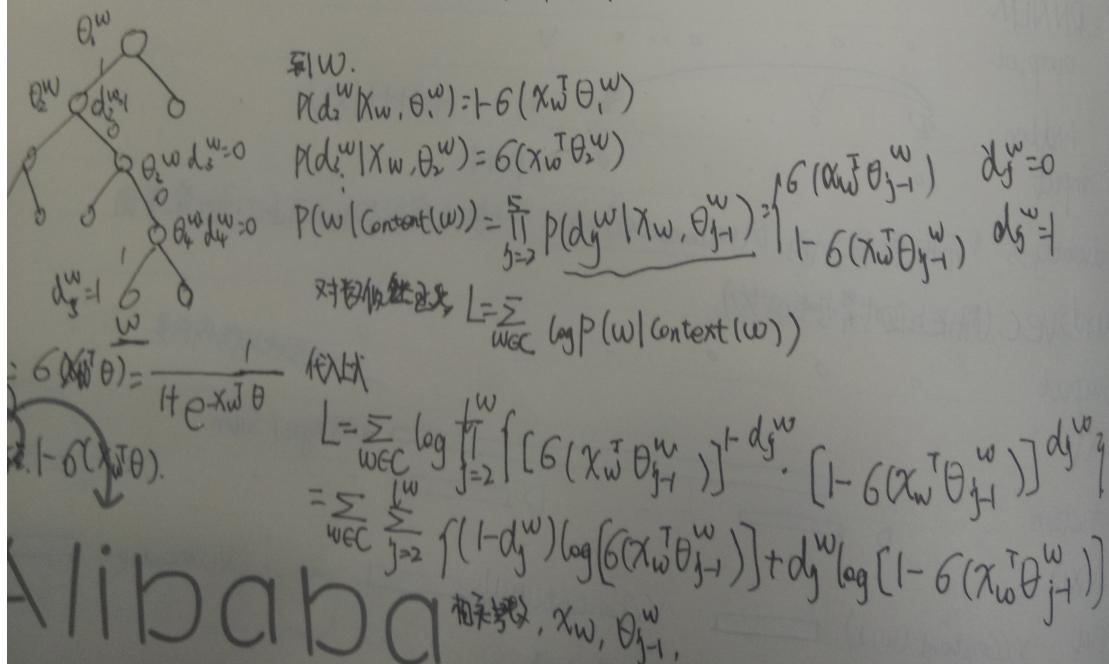
梯度计算： $\frac{\partial L}{\partial w}$ ：从根节点到 w 对应叶节点的路径。叶节点的梯度向量还是输入是同向量

$\frac{\partial L}{\partial w}$ ：路径 $P(w)$ 中包含的所有边。

P_1, P_2, \dots, P_m ：路径 $P(w)$ 中的 m 个节点。

d_1, d_2, \dots, d_m ：词 w 的 Huffman 编码，由 $L^W - 128, 128$ 成。 d_j 指射门的编码。

$\theta_1, \theta_2, \dots, \theta_m \in R^m$ ：路径 $P(w)$ 中非叶节点对应的向量。



④ 二层神经网络 (内线) : 1818
 $\frac{\partial L(w, j)}{\partial \theta_{j+1}^w} = [1 - d_j^w - \sigma(x_w^\top \theta_{j+1}^w)] x_w$ 梯度下降法
 $\Rightarrow \theta_{j+1} := \theta_{j+1} + \eta [1 - d_j^w - \sigma(x_w^\top \theta_{j+1}^w)] x_w$ (j是目标)
 $\frac{\partial L(w, j)}{\partial x_w} = [1 - d_j^w - \sigma(x_w^\top \theta_{j+1}^w)] \theta_{j+1}^w$
 $x_w \in \text{Content}(w)$ 中向量的权重。
 $V(\tilde{w}) = V(w) + \eta \sum_{j=2}^W \frac{\partial L(w, j)}{\partial x_w}$ $w \in \text{Content}(w)$
贡献于该向量上

CBOW 模型伪代码。
 $e = 0$ V 是最前面的词向量。
 $x_w = \sum_{u \in \text{Content}(w)} V(u)$
 For $j=2 \leq w$ do 遍历所有样本 $(\text{content}(w), w)$
 $\quad q = \sigma(x_w^\top \theta_{j+1}^w)$ 更新所有层中的参数。
 $\quad g = \eta(1 - d_j^w - q)$
 $\quad e = e + g \theta_{j+1}^w \Rightarrow \left(\eta \frac{\partial L(w, j)}{\partial x_w} \right)$.
 $\quad \theta_{j+1}^w = \theta_{j+1}^w + g x_w$
y.

for $u \in \text{Content}(w)$ do
 $V(u) = V(u) + e$

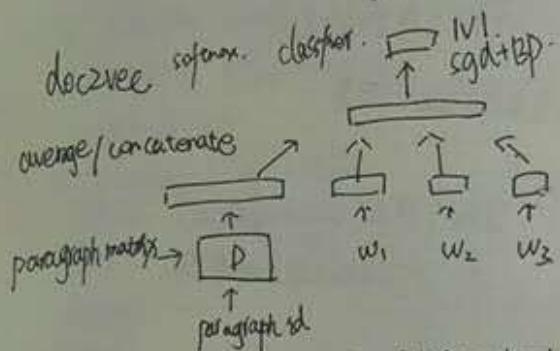
(2) Skip-gram. 已知上下文 $\text{Content}(w)$ 中的词向量。
 input-layer $V(w)$. $C \times D$ 特征矩阵
 projection: $V(w) \rightarrow$ $C \times D \times \log_2 V$ $P(u|w) = \prod_{j=1}^D P(d_j^u | V(w), \theta_{j+1}^u)$
 output $L = \sum_{w \in C} \sum_{u \in \text{Content}(w)} \sum_{j=1}^W \{ (1 - d_j^u) \cdot \log(\sigma(V(w)^\top \theta_{j+1}^u)) + d_j^u \cdot \log(1 - \sigma(V(w)^\top \theta_{j+1}^u)) \}$
决策树。
 $\Rightarrow \theta_{j+1}^u := \theta_{j+1}^u + \eta [1 - d_j^u - \sigma(V(w)^\top \theta_{j+1}^u)] V(w)$
 $\frac{\partial L(w, u, j)}{\partial V(w)} \Rightarrow V(w) = V(w) + \eta \sum_{u \in \text{Content}(w)} \sum_{j=1}^W \frac{\partial L(w, u, j)}{\partial V(w)}$
Word2Vec 中每次只取 $\text{Content}(w)$ 中一个， $\Pi - V(w)$ 。
 for $u \in \text{Content}(w)$ do
 $\quad e = 0$
 \quad for $j=2 \leq w$ do
 $\quad \quad q = \sigma(V(w)^\top \theta_{j+1}^u)$ $e = e + g \theta_{j+1}^u$
 $\quad \quad g = \eta(1 - d_j^u - q)$ $\theta_{j+1}^u = \theta_{j+1}^u + g V(w)$
 $\quad V(w) = V(w) + e$

重复等工强调。

Sub-Sampling. $\text{prob}(w) = \frac{1}{N} f(w)$, $f(w)$ 为 w 的频率

$$f(w) = \frac{\sum_{w \in D} \text{count}(w)}{|V|}$$

$$\text{frequency } f(w) = D_0 \left(1 - \frac{w - \text{actual}}{\text{total words} + 1} \right) \quad \text{当出现词频越高时, } f \text{ 越小.}$$



从一个随机的 paragraph 和样本文本的 content, 计算梯度并更新.

对同一个 paragraph, paragraph vector 表示, 只在训练期间使用, 用整个句子的语义.

在测试阶段, 对一个 paragraph, 给一个 paragraph id, 利用质心叫法的 word vectors 和神经网络模型, 用梯度下降训练将训练句子收敛到句子的 paragraph vector.

推导中 X2vec

$$P(W_t = v_i | U, C) = \frac{e^{Uv_i}}{\sum_j e^{Uv_j}} \rightarrow \text{Negative Sampling} \quad (\text{Hierarchical Softmax})$$

质心 distance (U_i, V_j) nearest top N

Alibaba

