Word2vec 原理解析

1. 前言

近年来，以深度学习为代表的人工智能技术发展如火如荼，其应用领域也从最开始的计算机视觉扩展到了包括自然语言处理、推荐、语音识别等在内的众多研究方向，并且在大多数领域取得了令人惊艳的效果，以笔者熟悉的自然语言处理（NLP）方向为例，纵观近几年在ACL、EMNLP、CLONG等顶级学术会议上发表的学术论文，随处可见Deep Learning、Neural Network/NN、RNN、Convolutional Neural Networks/CNN、LSTM、Autoencoder、Embedding等字样，传统的自然语言处理的任务比如文本分类、情感分析、命名实体识别、分词、关系抽取、机器翻译等任务，目前取得State of Art 的方案也基本被基于深度学习的算法占FF据。文本表示是NLP中的基础，Word2Vec作为一种基于神经网络思想的文本表示学习方法，对在深度学习盛行的今天想要学习NLP的人来说是必要的入门知识。

word2vec是谷歌开源的一款用于词向量计算的工具，同时也是一套生成词向量的算法方案，其他生成词向量的方案有LSA、PLSA、LDA等等。Word2Vec算法的背后是一个浅层神经网络，其网络深度仅为3层，所以严格说Word2Vec并非深度学习范畴，但其生成的词向量在很多任务中都可以作为深度学习算法的输入，因此在一定程度上可以说Word2Vec技术是深度学习在NLP领域能够更好应用的基础。

本文主要涉及一下知识点：

* 1) 相关背景知识 * 2) Word2Vec中两种词向量模型CBOW和Skip-gram * 3) Word2Vec中两种学习框架Hierarchical Softmax和Negative Samping的原理分析 * 4) Word2Vec相关拓展与应用

###2. 相关背景知识

####2.1 词向量

计算机不能直接处理文本、图像、声音等内容，需要将其转化为数字特征后才能处理，词向量就是文本中的单词转后的计算机能够处理的数字化特征。词向量的表示方式主要有两种，一种是one-hot representation，就是利用一个高维稀疏（所谓稀疏是指向量中为0的维度特别多）的向量表示一个单词，向量的维度为词典中词的个数，向量中只有一个维度的值为1，其余维度值为0，为1的维度为词在词典中的索引位置，常见的有词袋（BOW，bag of words）模型；另一种是distributed representation，该方法将单词表示成一个低维稠密的向量，常见的方法有LSA、PLSA、LDA、Word2vec、Glove等等。下面以单词powerful为例说明两种不同的表示方式：

S=(w1,w2,…,wT)

wi表示S中的第i的单词，则文本S出现的概率为：

![enter image description here](http://images.gitbook.cn/72096170-bd20-11e7-8439-65f318c92b02)

上面公式利用了贝叶斯公式进行了分解，上式中各个概率值是语言模型的参数，对于一段新给出的文本，只需将分解后的参数进行累乘即可得出新文本出现的概率。

语言模型参数的更一般表示形式为：p(w/context(w))，其中w表示某个单词，context(w)表示单词w的上下文，含义为L上下文为context(w)时,词w出现的概率。

学习语言模型参数p(w/context(w))，目前解决方案有很多，例如N-gram模型、N-pos模型和神经概率语言模型（NPLM）以及本文提到的Word2Vec中的Hierarchical Softmax模型和Negative Sampling模型（其实本质上HS和Negative Sampling是对传统NPLM语言模型的一种简化，其中基于HS的实现是利用Huffman Tree替代了原有的全连接网络）。

这里特别说明一点：词向量是神经网络统计语言模型的副产品。

###3. CBOW模型

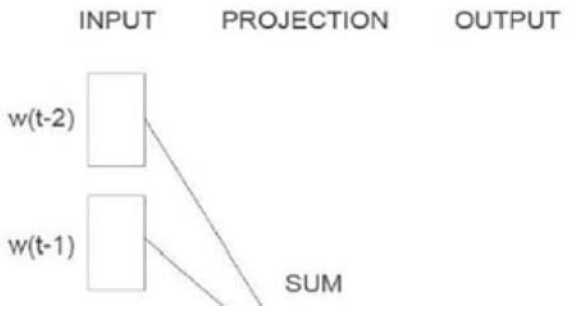
下面介绍连续词袋模型（CBOW model ,Continuous Bag of Words Model）。

####3.1 模型结构

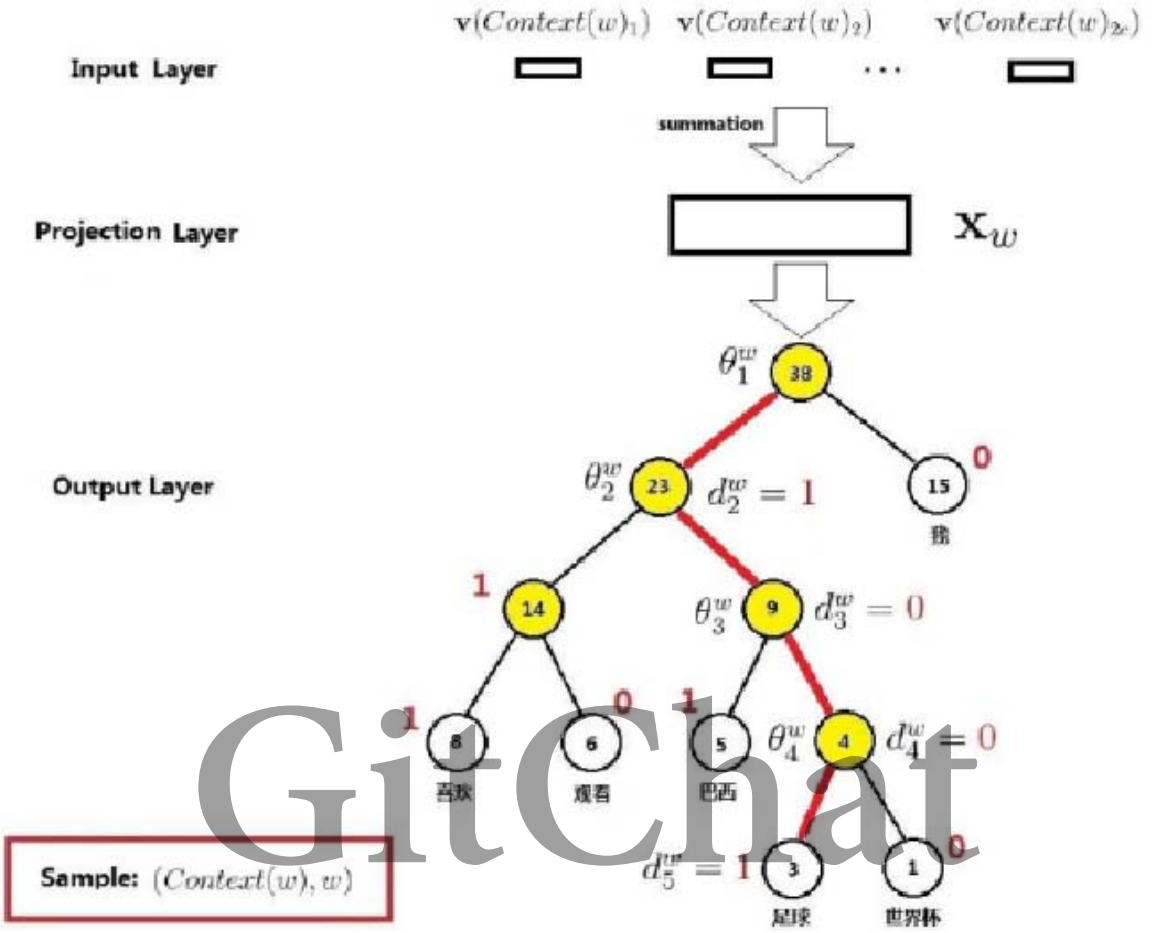
模型包含三层，输入层，映射层和输出层。该网络的含义是已知词w(t)的上下文w(t-2)，w(t-1)，w(t+1)，w(t+2)的前提下预测词w(t)出现的概率，即：p(w/context(w))。 目标函数为：

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

CBOW 模型中p(w/context(w))的表示框架有两种，分别为：Hierarchical Softmax和Negative Samping，下面分别对两种框架下的模型参数求解进行介绍。



基于Hierarchical Softmax框架的CBOW模型将输出层利用一棵Huffman Tree进行表示，该Huffman Tree使用语料中的词根据词频进行构建，树中每个叶子节点代表一个单词，约定（词频较大的）左孩子节点编码为1，（词频较小的）右孩子节点编码为0。同时利用节点的层级关系，将从根节点到单词t的路径中每个节点当做一次二分类（基于Logistic Regression算法），将整个路径产生的概率作为p(w/context(w))。



网络包含以下几部分：

- 1) 输入层：包含Context(w)中2c个词向量v(Context(w)1), v(Context(w)2),…v(Context(w)2c),每个词向量的维度为m。
- 2) 投影层：将输入层的2c个向量求和累加，即：

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

对节点进行分类使用的是逻辑回归算法，因此一个节点被分为正类的概率是

$$\sigma(\mathbf{x}_w^\top \theta) = \frac{1}{1+e^{-\mathbf{x}_w^\top \theta}},$$

被分为负类的概率为

$$1-\sigma(\mathbf{x}_w^\top \theta),$$

其中θ为待定参数，每个非叶节点对应一个，设设为$_i^w$，对于从根节点出发到达”足球”叶节点共经历了4次二分类，每次分类结果的概率为：

$$\begin{aligned}
&1.\ \text{第 1 次:}\quad p(d_2^w|\mathbf{x}_w,\theta_1^w) = 1-\sigma(\mathbf{x}_w^\top\theta_1^w);\\
&2.\ \text{第 2 次:}\quad p(d_3^w|\mathbf{x}_w,\theta_2^w) = \sigma(\mathbf{x}_w^\top\theta_2^w);\\
&3.\ \text{第 3 次:}\quad p(d_4^w|\mathbf{x}_w,\theta_3^w) = \sigma(\mathbf{x}_w^\top\theta_3^w));\\
&4.\ \text{第 4 次:}\quad p(d_5^w|\mathbf{x}_w,\theta_4^w) = 1-\sigma(\mathbf{x}_w^\top\theta_4^w),
\end{aligned}$$

p(足球/Context(足球))跟这四个概率值的关系有：

$$p(\text{足球}|Context(\text{足球})) = \prod_{j=2}^{5} p(d_j^w|\mathbf{x}_w,\theta_{j-1}^w).$$

综上所述：对于训练语料的词典D中任意次w,Huffman树上必然存在唯一一条从根节点到词w对应节点的路径p^w,路径p^w上存在l^{w-1}个分支，将每个分支看做一次二分类，每一次二分类产生一个概率，这些概率的乘机即为条件概率p(w/context(w))，其一般形式如下：

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

其中:

$$\begin{aligned}\mathcal{L} &= \sum_{w\in\mathcal{C}} \log \prod_{j=2}^{l^w} \left\{ [\sigma(\mathbf{x}_w^\top \theta_{j-1}^w)]^{1-d_j^w} \cdot [1-\sigma(\mathbf{x}_w^\top \theta_{j-1}^w)]^{d_j^w} \right\} \\ &= \sum_{w\in\mathcal{C}} \sum_{j=2}^{l^w} \left\{ (1-d_j^w) \cdot \log[\sigma(\mathbf{x}_w^\top \theta_{j-1}^w)] + d_j^w \cdot \log[1-\sigma(\mathbf{x}_w^\top \theta_{j-1}^w)] \right\},\end{aligned}$$

接下来对L求极大值，需要对L中的变量(xw和θ_(j-1)^w)进行求解梯度，为方便梯度推导令：

$$\mathcal{L}(w,j) = (1-d_j^w) \cdot \log[\sigma(\mathbf{x}_w^\top \theta_{j-1}^w)] + d_j^w \cdot \log[1-\sigma(\mathbf{x}_w^\top \theta_{j-1}^w)].$$

首先计算L(w,j)关于$\theta_{(j-1)}^w$的梯度：

$$\frac{\partial \mathrm{L}(w,j)}{\partial \vartheta_{j-1}^w} = \frac{\partial}{\partial \vartheta_{j-1}^w}\left\{\left(1-\mathrm{d}_{j-1}^w\right)\cdot \log\left[\sigma\left(x_w^T \vartheta_{j-1}^w\right)\right] + \mathrm{d}_j^w \cdot \log\left[1-\sigma\left(x_w^T \vartheta_{j-1}^w\right)\right]\right\}$$

$$= \left(1-\mathrm{d}_j^w\right)\left[1-\sigma\left(x_w^T \vartheta_{j-1}^w\right)\right] x_w - \mathrm{d}_j^w \sigma\left(x_w^T \vartheta_{j-1}^w\right) x_w$$

$$= \left\{\left(1-\mathrm{d}_j^w\right)\left[1-\sigma\left(x_w^T \vartheta_{j-1}^w\right)\right] x_w - \mathrm{d}_j^w \sigma\left(x_w^T \vartheta_{j-1}^w\right)\right\} x_w$$

$$= \left(1-\mathrm{d}_j^w - \sigma\left(x_w^T \vartheta_{j-1}^w\right)\right) x_w$$

其中η为更新速率。

同样的，L(w,j)关于x_^w的梯度为：

$$\vartheta_{j-1}^w := \vartheta_{j-1}^w + \eta\left[1-\mathrm{d}_j^w - \sigma\left(x_w^T \vartheta_{j-1}^w\right) x_w\right]$$

因此，x_w的更新公式为：

$$\mathbf{v}(\widetilde{w}) := \mathbf{v}(\widetilde{w}) + \eta \sum_{j=2}^{l^w} \frac{\partial \mathcal{L}(w,j)}{\partial \mathbf{x}_w}, \quad \widetilde{w} \in Context(w),$$

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

表示w的标签，即正样本标签为1，负样本标签为0，对比HS模型中的非叶子节点的分类。

对于给定的词w和上下文Context(w)，我们的目的是最大化下式：

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

其中：

$$p(\mu|Context(\omega)) = \begin{cases} \sigma\left(x_w^T \vartheta_{j-1}^w\right), & L^\omega(\mu) = 1 \\ 1 - \sigma\left(x_w^T \vartheta_{j-1}^w\right), & L^\omega(\mu) = 0 \end{cases}$$

或者整体表达为：

$$p(\mu|Context(\omega)) = [\sigma\left(x_w^T \vartheta_{j-1}^w\right)]^{L^\omega(\mu)} \cdot [1 - \sigma\left(x_w^T \vartheta_{j-1}^w\right)]^{1-L^\omega(\mu)}$$

对于一个给定的语料库C，模型的目标函数为：

$$G = \prod_{w \in \mathcal{C}} g(w)$$

为了方便计算，对其取对数操作，即为L

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

$$\frac{\partial \mathcal{L}(w,u)}{\partial \mathbf{x}_w} = \left[L^w(u) - \sigma(\mathbf{x}_w^\top \theta^u)\right] \theta^u.$$

因此，θ^u 和x_w的更新公式为：

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

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

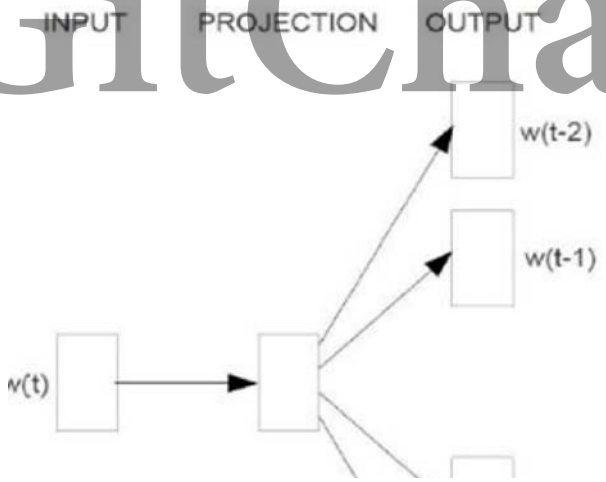
4. Skip-gram模型

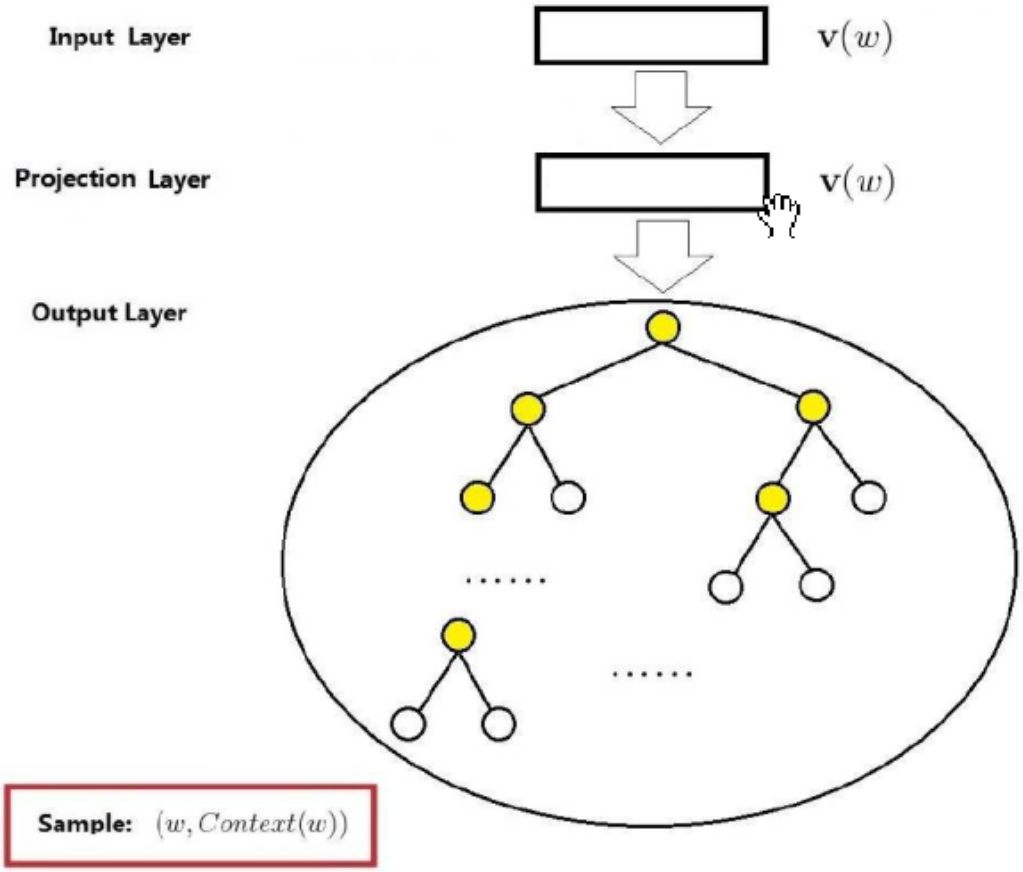
4.1 模型结构

模型同样包含三层，输入层，映射层和输出层。该网络的含义是已知词w(t)的前提下预测词w(t)的上下文w(t-2)，w(t-1)，w(t+1)，w(t+2)，条件概率写为：p(context(w)/w)。

目标函数为：

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





基于Hierarchical Softmax框架的Skip-gram模型同样将输出层利用一棵Huffman Tree进行表示，该Huffman Tree使用语料中的所有词根据词频进行构建，树中每个叶子节点代表一个单词，同时利用节点的层级关系，将从根节点到context(t)中每个单词所在叶子节点的路径中每个节点当做一次二分类（基于Logistic Regression算法），将所有路径产生的概率作为p(w/ context(w))。

- 1)输入层：包含1个词向量v(w)，词向量的维度为m。
- 2)投影层：与输入层相同
- 3)输出层：输出层对应一棵二叉树，它是以语料中快出现过词作为节点，以各词在语料中出现的次数当权值构造出来的Huffman树。这个Huffman树中，叶子节点功N个，分别对应词典中的词，非叶子节点N-1个（上图中黄色结点）。

在skip-gram模型中p(w/ context(w))的定义为：

将上面公式带入目标函数

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

得：

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

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

下面使用梯度上升对L进行优化，分别对公式中的变量求偏导数：

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

$$\frac{\partial \mathcal{L}(w,u,j)}{\partial \mathbf{v}(w)} = \left[1 - d_j^u - \sigma(\mathbf{v}(w)^\top \theta_{j-1}^u)\right] \theta_{j-1}^u.$$

因此，$\theta_{(j-1)}^{u}$的更新公式为：

$$\theta_{j-1}^u := \theta_{j-1}^u + \eta \left[1 - d_j^u - \sigma(\mathbf{v}(w)^\top \theta_{j-1}^u)\right] \mathbf{v}(w)$$

v(w)的更新公式为：

$$\mathbf{v}(w) := \mathbf{v}(w) + \eta \sum_{u \in Context(w)} \sum_{j=2}^{l^u} \frac{\partial \mathcal{L}(w,u,j)}{\partial \mathbf{v}(w)}.$$

NEG(u)表示处理词u时生成的负样本子集，条件概率

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

写成完整的表达式

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

为了简化运算，最终目标函数取对数形式：

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

下面对公式中的变量求偏导，进而得出变量的更新公式，具体推导过程参考之前的方法，这里不再赘述。

5. 相关应用与拓展

5.1 Word2Vec的应用

1) 近义词/相似文本发现

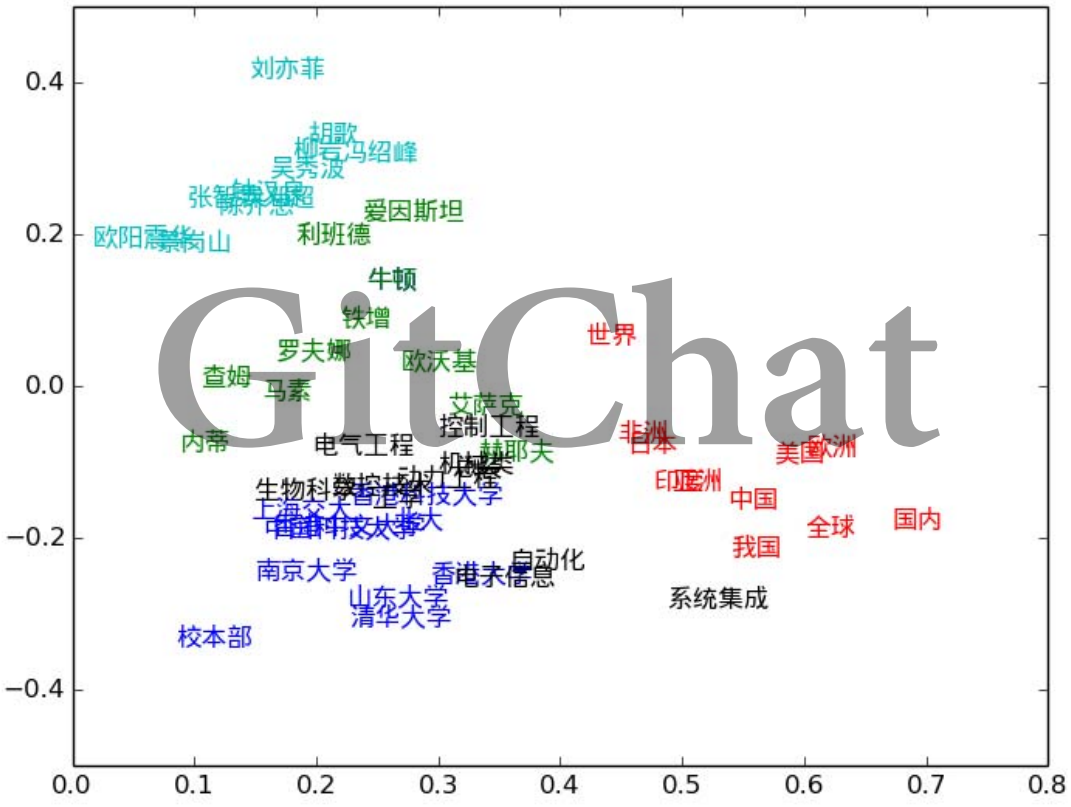
```
请输入词语<exit退出>:中国
大陆      0.66763467
中共      0.57856727
共产党    0.56305367
解放军    0.55761635
台湾      0.5368497
反攻      0.5271177
日本      0.5103535
王文莹    0.49295437
内地      0.48557448
对岸      0.48428434
```

```
请输入词语<exit退出>:钓鱼岛
钓鱼台    0.6219264
钓岛      0.6123347
南海      0.6018163
领土      0.51753837
领海      0.4928774
岛屿      0.4853142
舰队      0.47854927
渔权      0.47229362
主权      0.46729872
东海      0.4613399
```

```
请输入词语<exit退出>:旅游
观光      0.65619475
景点      0.60212
陆客      0.59477097
旅行      0.5677106
游憩      0.557839
赏樱      0.5571045
游玩      0.52199984
观光客    0.51974636
行程      0.51943743
参观      0.5077874
```

```
请输入词语<exit退出>:苹果
三星      0.7224437
微软      0.7101249
Apple     0.66682446
iPhone5   0.62071097
Google    0.597368
iPadmini          0.5609188
新机      0.559093
库克      0.5589176
宏达电    0.555409
产品      0.55437565
```

下图为使用PCA将词向量进行降维后的效果展示：



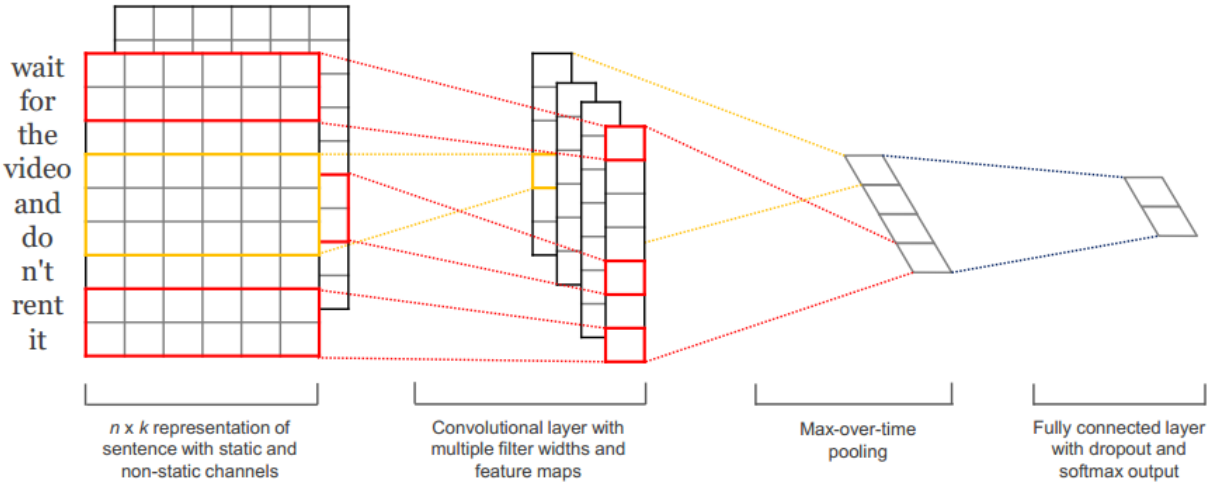
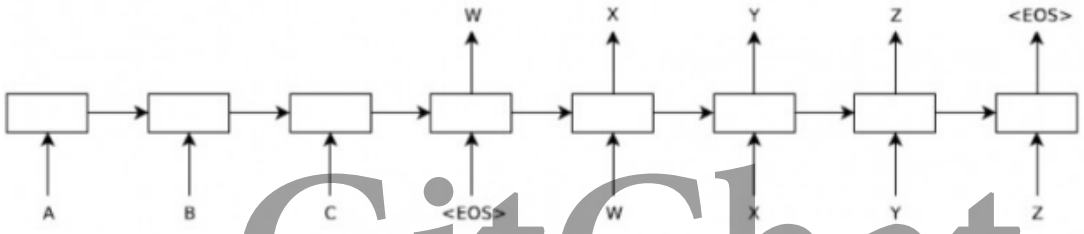
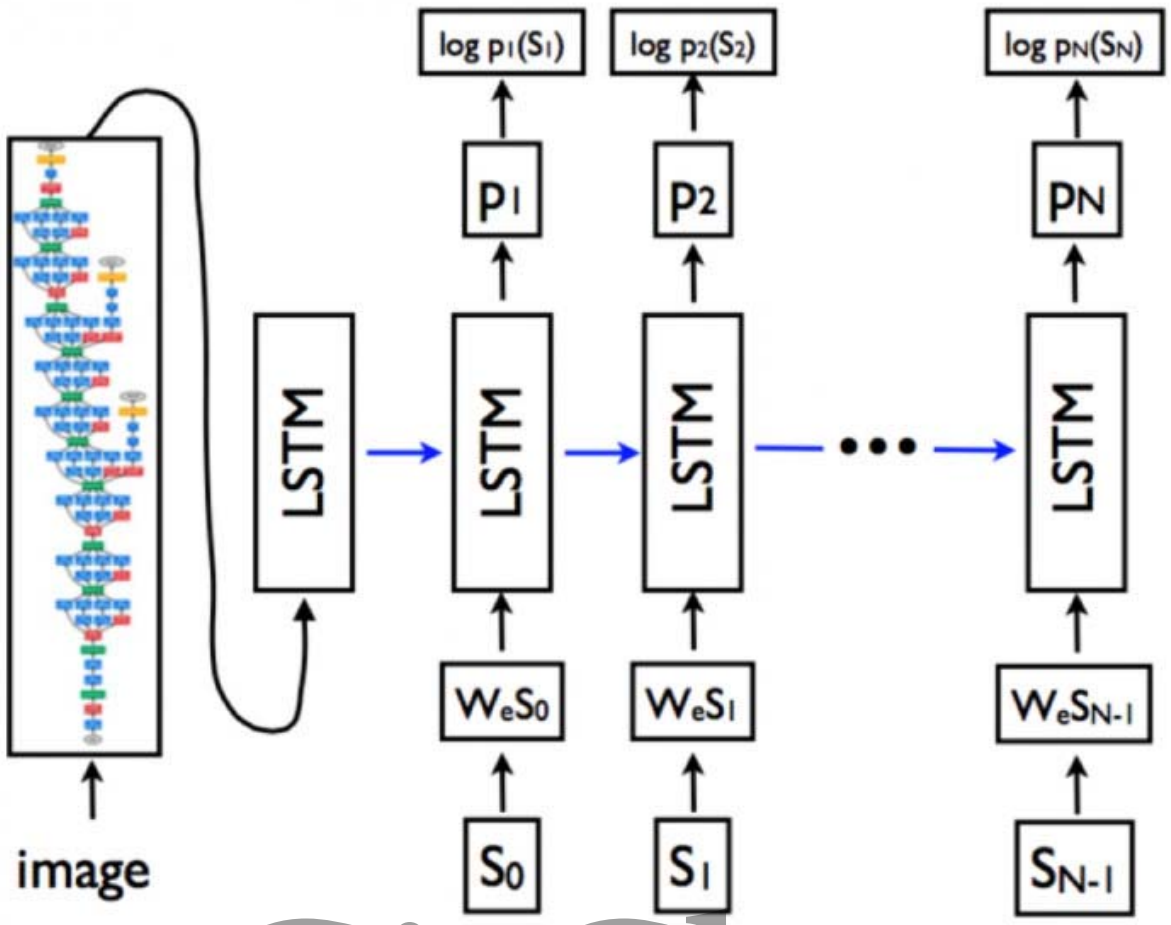


Figure 1: Model architecture with two channels for an example sentence.

同样的，在分词、实体抽取、机器对话、神经网络机器翻译（NMT）等任务中广泛应用的Seq2Seq模型中也有word2vec的应用，类似的词向量也是作为词的特征的进行输入。

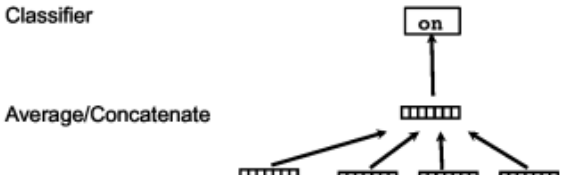


类似的应用还有生成自动图像描述等，模型网络结构如下图：



5.2 其他知识表示

归根到底，word2vec是一种对单词知识的表示方法，word2vec之后出现了大量知识表示方法的研究，例如“段向量”（paragraph vector），提出者将段信息作为一个特殊的词，参与模型的训练，模型结构基本与word2vec相同，具体如下图所示：



类似的知识表示还有graph2vec、thought2vec等等。

6. 参考文献

- [1] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.
- [2] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013: 3111-3119.
- [3] Le Q, Mikolov T. Distributed representations of sentences and documents[J]. 2014, 4:II-1188.
- [4] A. Minnaar, “Part II - Continuous Bag-of-Words Model,” 12 4 2015. [Online]. Available:http://mccormickml.com/assets/word2vec/Alex_Minnaar_Word2Vec_Tutorial_Part_I_The_Skip-Gram_Model.pdf.
- [5] C. McCormick, “Word2Vec Tutorial Part 2 - Negative Sampling,” 11 1 2017. [Online]. Available: http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negativesampling/.
- [6] C. McCormick, “Word2Vec Tutorial - The Skip-Gram Model,” 19 4 2016. [Online]. Available: http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-grammodel/.
- [7] peghoty,”word2vec 中的数学,”2014,pp.19-20.