# 作业二：词向量训练 实验报告

**陈婧明 202318019427004**

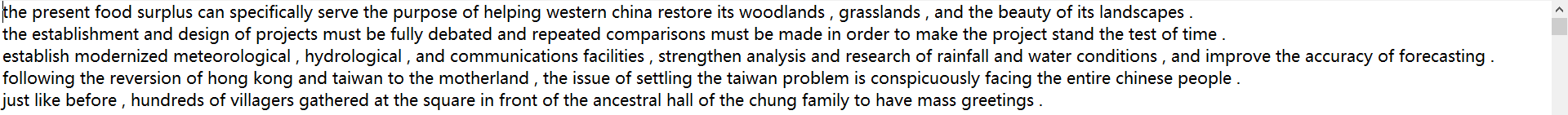
## 训练方式

实验代码用 Python 语言编写，在 PyCharm 2022 平台上开发。

本机上的实验使用的深度学习框架是 PyTorch 1.8.2，显卡为 NVIDIA GeForce RTX 3050，CUDA 版本为 11.4。

在colab平台上实验使用PyTorch 2.0.1，显卡Tesla T4，CUDA 版本为 11.8。

数据：中文、英文语料各一份，数据已预先分词，有标点符号，如图。





实验主要参考论文中的负采样方法训练词向量。

distributed-representations-of-words-and-phrases-and-their-compositionality

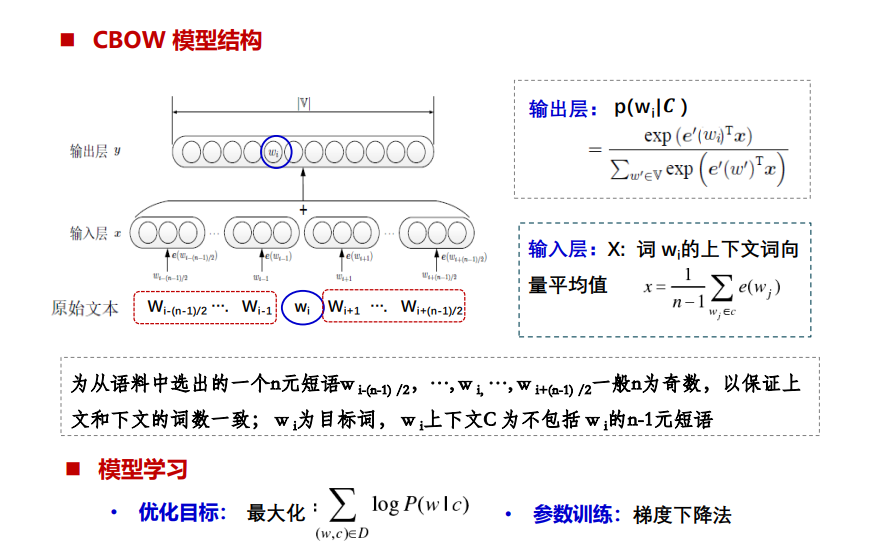
## 模型结构

* + - 1. **模型介绍**

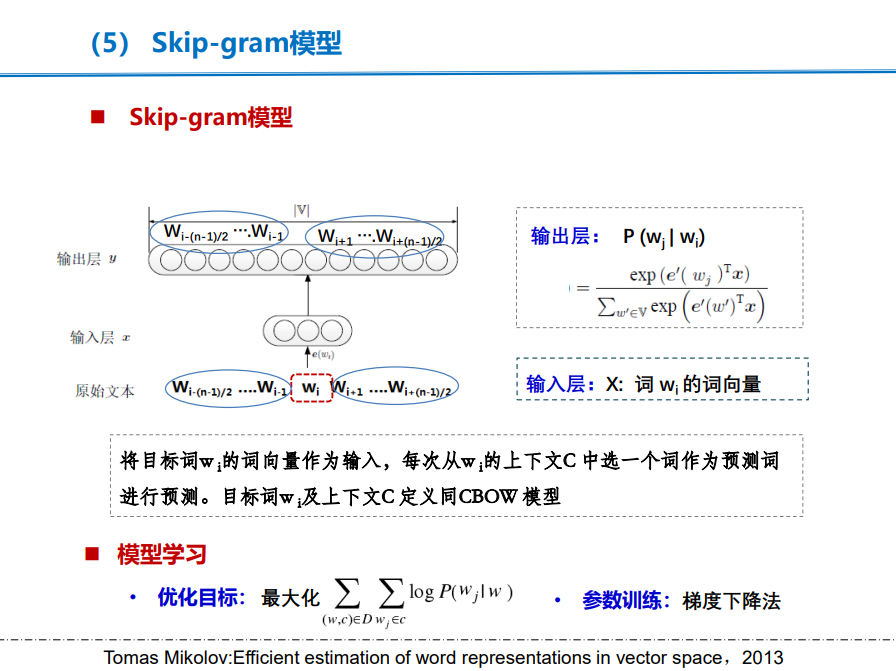
**词向量：**使用one-hot的方式表示每个词，需要建立一个和词典一样大的向量，然后词典位置用1，其他位置用0。但是，这种表示方法把每个词孤立起来了，使得算法对相关词的泛化能力不强。

**Word-Embedding：**事先训练好一个词嵌入矩阵 embedding matrix，他的每一列对应每个单词的词向量，每一行表示一个特征。例如，一个100\*10000的矩阵，就是对应10000个单词，每个单词都是从100个特征上进行衡量。这样，我们拿这个矩阵乘以每个单词的one-hot的表示，就会得到每个单词的词向量表示，构造低维稠密向量作为词的分布式表示。

**Skip-Gram模型：**拥有相似或者相同的上下文的多个词可能是近义词或者同义词，因此，抽取上下文和目标词配对，构造一个监督学习问题是一种训练词向量的有效方法。Skip-Gram模型是Word2Vec的一种，通过目标词预测上下文，而Word2Vec的另一种模型CBOW是通过上下文去预测目标词。



CBOW模型



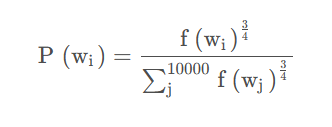
Skip-gram模型

**Negative sampling：**是另外一种有效的求解embedding matrix的方法。它的做法是判断选取的context word和target word是否构成一组正确的context-target对，一般包含一个正样本和k个负样本。

例如，“orange”为context word，“juice”为target word，很明显“orange juice”是一组context-target对，为正样本，相应的target label为1。若“orange”为context word不变，target word随机选择“king”、“book”、“the”或者“of”等。这些都不是正确的context-target对，为负样本，相应的target label为0。

生成训练集时，选1个正样本对应K个负样本（样本是成对出现的），一般地，若训练样本较小，k一般选择5～20；若训练样本较大，k一般选择2～5即可。如果词表有10000个单词，之前的方法需要10000个输出单元的softmax分类，而negative sampling转化为k+1个二分类问题，计算量要小很多，大大提高了模型运算速度。

正样本是该目标词的上下文，关于如何选择负样本，论文中提出一个更实用、效果更好的方法，就是根据该词出现的频率进行选择，相应的概率公式为：

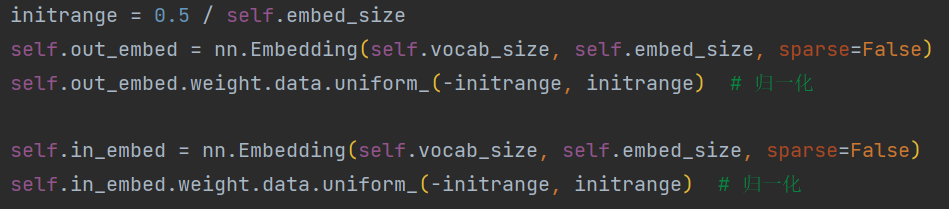


* + - 1. **网络结构**

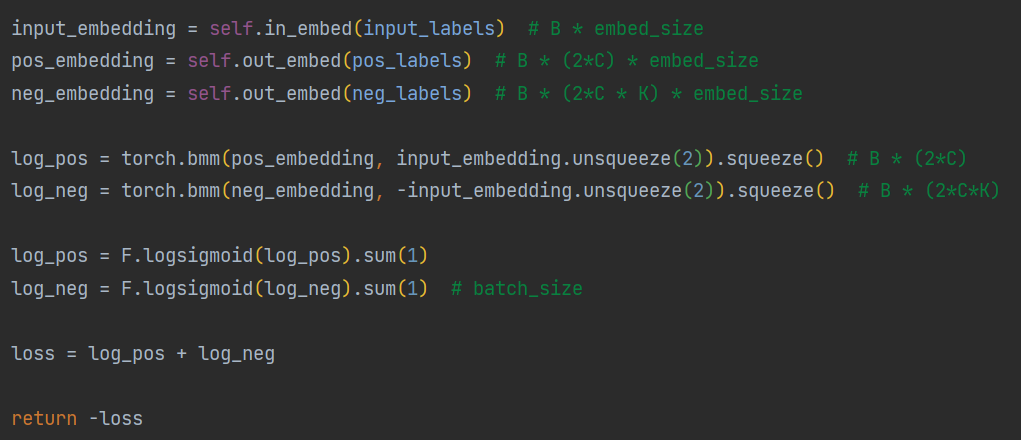
在pytorch中使用函数nn.Embedding(vocab\_size, embedding\_dim) 作为字典映射表，其中vocab\_size表示词表的大小，embedding\_dim表示词向量维度。

in\_embed：映射输入目标词，vocab\_size = 10000，embedding\_dim = 100。

in\_embed：映射输入目标词，vocab\_size = 10000，embedding\_dim = 100。

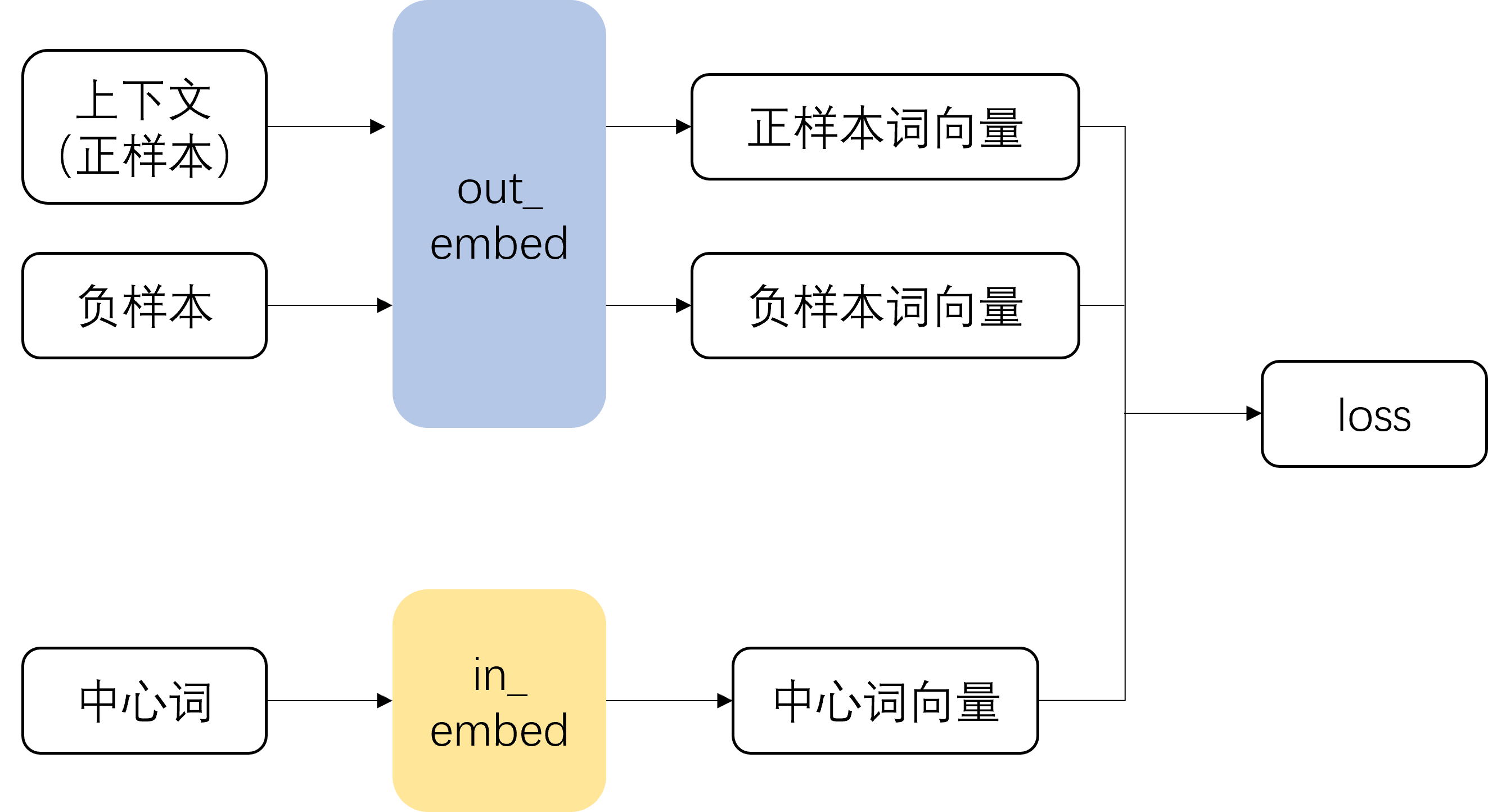


模型结构



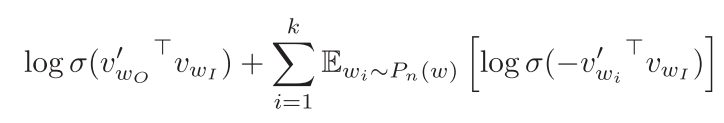
Forward

网络结构如图所示。



网络结构

* + - 1. **损失函数**

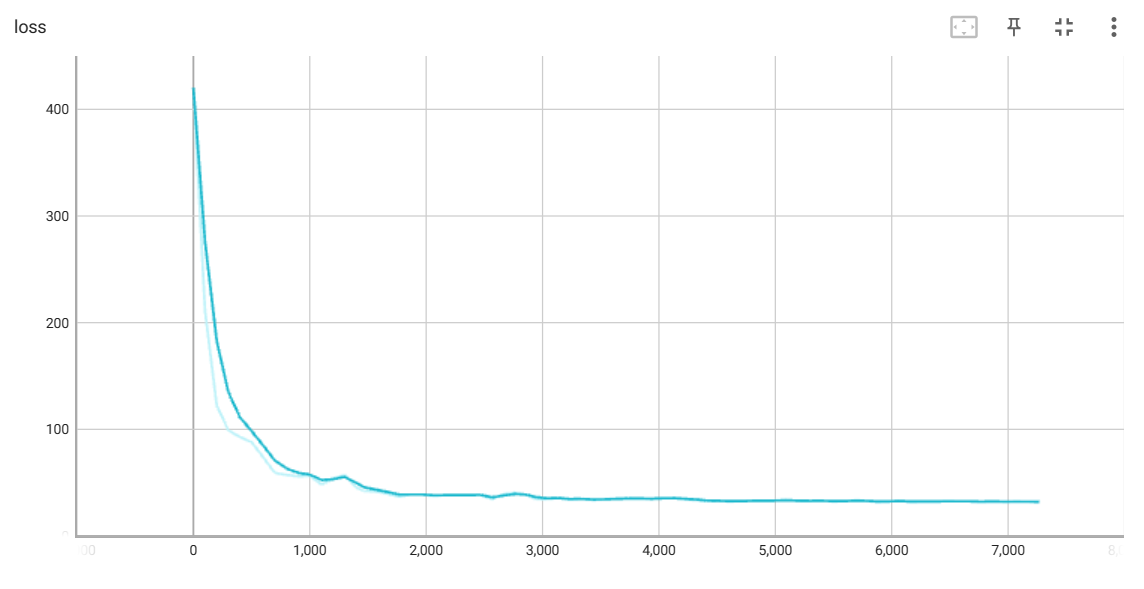


前面的部分是正样本和context的关系，后面部分是负样本和context的关系。如果两个向量的关系越接近，那么内积就会越大。我们希望的是正样本与context的关系尽可能的近，也就是前面那部分越大越好，希望负样本与context的关系尽可能的小，但是后面发现内积前加了个负号，那就表示后面那部分越大越好。所以损失函数应该越大越好。编程中最小化损失函数的相反数 -loss。

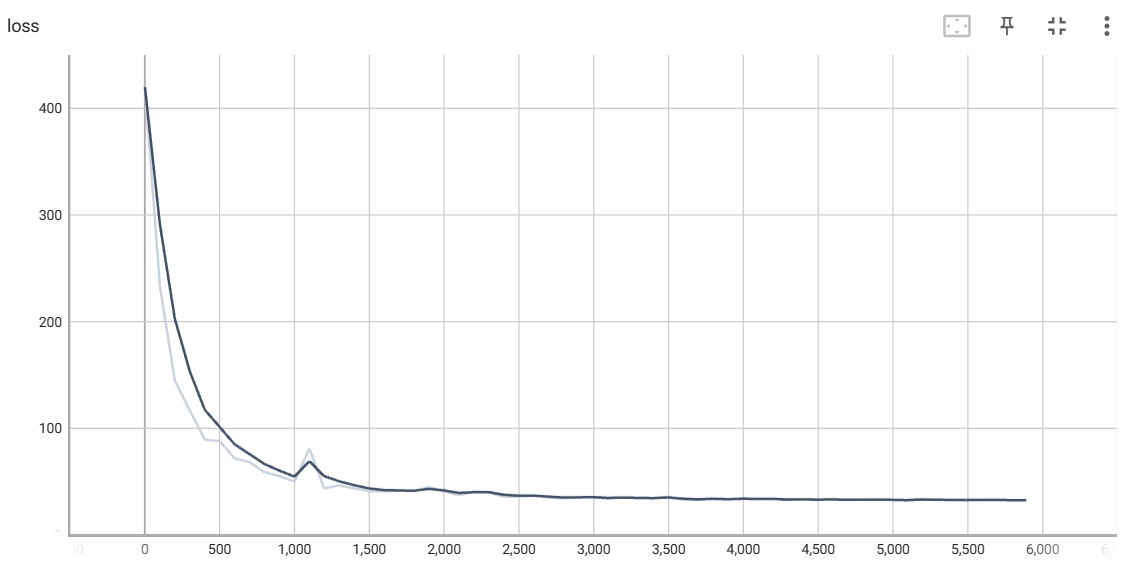
## 实验结果

* + - 1. **损失函数**

损失函数下降趋势如图所示。可以看出模型收敛速度很快，训练到2000轮后损失函数基本不再下降。



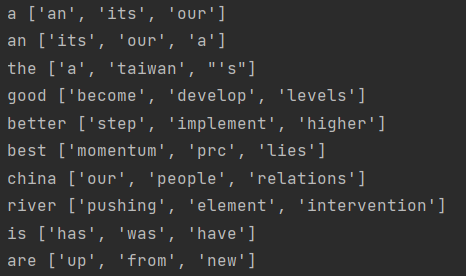
英文语料loss



中文语料loss

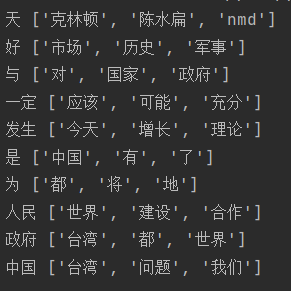
* + - 1. **英文近义词**

寻找 ["a", "an", "the", "good", "better", "best", "china", "river", "is", "are"] 的近义词，可以看出 ["a", "an", "the"] 的近义词多为冠词或所有格形式，因为他们都可以用在名词的前面，而形容词 ["good", "better", "best"] 和名词 ["china", "river"] 的近义词训练效果不是很理想，但是仍然能找到 better 的近义词 higher，china 或river 的近义词多为名词。



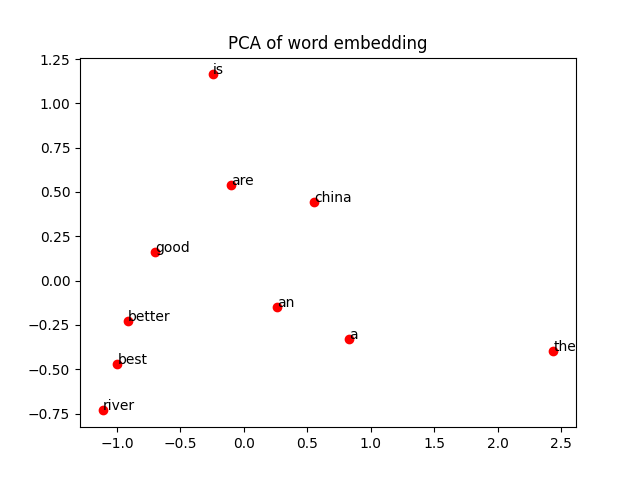
* + - 1. **中文近义词**

寻找 ["自由", "好", "与", "一定", "发生", "是", "为", "人民", "政府", "中国"] 的近义词，可以看出名词的近义词多为名词，副词的近义词多为副词，动词的近义词多为动词。 比如，"一定" 的近义词都是副词 '应该', '可能', '充分'，"中国" 的近义词都是名词 '台湾', '问题', '我们'。

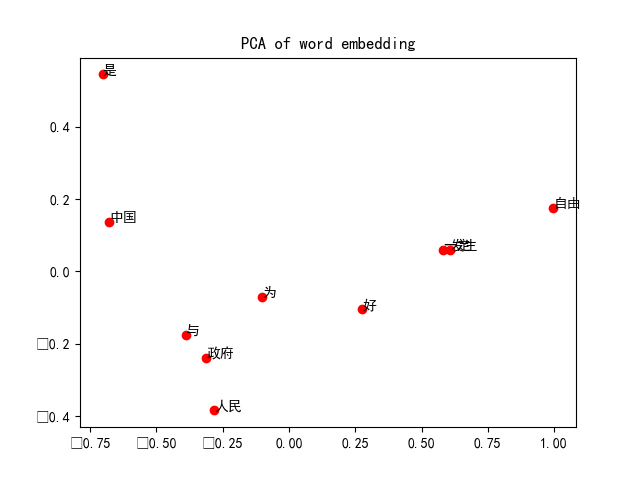


* + - 1. **可视化**

训练好的词向量为100维，为了将词向量进行可视化，用PCA降维方法将维度降到2维，在平面坐标系中进行表示。可视化结果如图所示，仍然取上述的几个单词（或词组）。



英文可视化结果



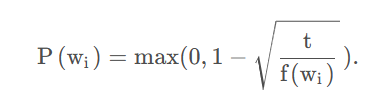
中文可视化结果

## 问题思考

* 1. **Subsampling：**如何处理标点符号，和“a、and、the”等在语料库中出现频率高，但是不太重要的词。

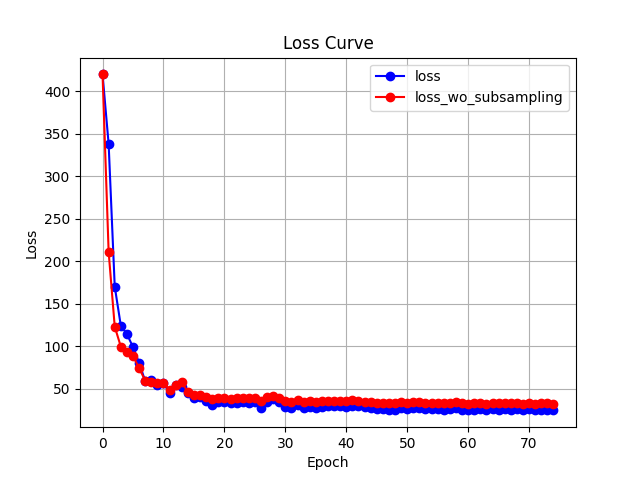
例如，虽然Skip-gram模型能够从“France”和“Paris”的关系中获益，但比它从“France”和“the”中获益要少得多，因为几乎每个单词都在一个带有“the”的句子中频繁地共同出现。

在上面的实验中没有处理上述问题，论文中提出了一种Subsampling of Frequent Words 的方法，使得低频词训练的机会变大，高频词训练机会降低，对于词，被剔除的概率为：



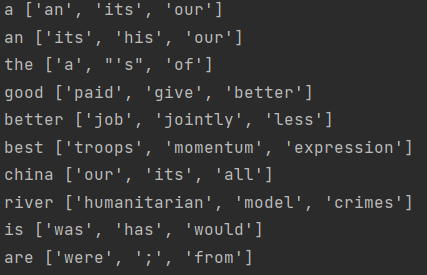
其中，是词出现的概率，t是人为设定的阈值（文中取）。文中提到，重采样加快了学习速度，甚至显著提高了学习生僻词向量的准确性。

只对英文语料进行subsampling，得到loss下降的曲线如图所示。可以看出添加subsampling后损失函数收敛到更小的值，但是可能是由于数据量有限，并没有看出收敛速度的优势。

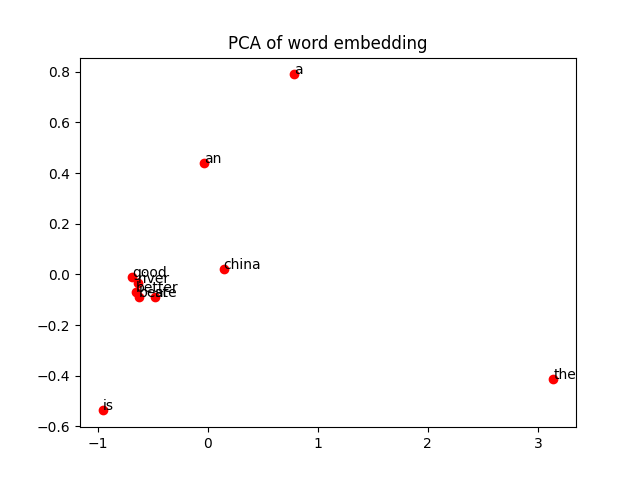


有无subsampling模块的loss曲线

选择10个单词的近义词和可视化的结果如下，相比于之前没有添加subsampling的实验结果，没有发现明显的改善。



添加subsampling的英文近义词



添加subsampling的英文可视化结果

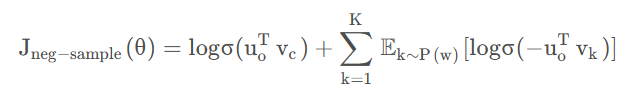
* 1. 我们最终希望得到的Look\_up 表（即embedding matrix）是一个矩阵，通过in\_embed.weight.data.cpu().numpy() 可以获取矩阵数据，并存储起来，而不必通过torch.save() 存储完整的网络模型和参数。



* 1. 为了将中文标签正确在figure中显示，需要加上一行代码：

plt.rcParams['font.sans-serif'] = ['SimHei']

* 1. CBOW模型的负样本：输入是窗口内**上下文词的平均**，正样本由中心词组成，负样本由随机采样的词组成。



* 1. word2vec

word2vec主要包括两种模型，分别是skip-gram和CBOW，其中是skip-gram是用中心词来预测周围词，而CBOW是用周围词来预测中心词。

论文对于skip-gram和CBOW模型有两种加速训练的方法，分别是层次softmax和负采样，层次softmax使用霍夫曼编码的二叉树进行展开，在语料库中出现频率大的词放在靠近根节点的叶子上，以便被快速找到。负采样将一个多分类的问题转化成二分类的问题，其正样本是一个中心词和他的周围词，负样本是一个中心词和随机采样的词。论文通过重采样（subsampling）技术把高频词删去一些，低频词尽量保留下来，这样可以加速模型训练，并且能够得到更好的词向量。